AAE

AERONAUTICAL AND ASTRONAUTICAL ENGINEERING DEPARTMENT

ENGINEERING EXPERIMENT STATION, COLLEGE OF ENGINEERING, UNIVERSITY OF ILLINOIS, URBANA

Aeronautical and Astronautical Engineering Department
University of Illinois Urbana, Illinois

Technical Report AAE 77-13

UILU-Eng 77-0513

NASA Grant NGR 14-005-194
Allen I. Ormsbee, Principal Investigator

PROPELLER STUDY PART II

THE DESIGN OF PROPELLERS FOR MINIMUM NOISE

by

Chung-Jin Woan

University of Illinois
Urbana, Illinois

July 1977

TABLE OF CONTENTS

INTRODUCTION

The trend in propeller aircraft has been toward increasing the speed, size, and horsepower. As a result, there is an increasing demand for the design of propellers which are efficient and yet produce minimum noise. This requires accurate determinations of both the flow over the propeller blade surfaces and the acoustic field induced by the moving propeller.

Much effort has been devoted recently to the development of a more sophisticated propeller theory. This theory has proceeded from the early simple momentum model of W. J. M. Rankine (1, 1865) and R. E. Froude (2, 1889) and the blade-element model of W. Froude (3, 1878) and S. Drzewiecki (4, 1920) to the vortex theory first proposed by F. W. Lanchester (5, 1907), the lifting-line model of S. Goldstein (6, 1929) and, finally, to the lifting surface model of H. Ludwieg and I. Ginzel (7, 1944). One of many important advancements in lifting-line theory is Lerbs' calculation of the induction factors (8, 1952), which allows the velocities at each blade section to be determined with great accuracy. This important calculation, plus other more sophisticated mathematical models, e.g., P. C. Pien (9, 1961), J. E. Kerwin (10, 1964), and W. B. Morgan (11, 1968), makes it possible today to design a propeller based on fluid dynamic principles.

One of the important problems of aeroacoustics is the determination of the sound from a rotating propeller. Historically, L. Gutin (12, 1936) was the first to theoretically investigate this sound for a static rotating propeller, using an equivalent distribution of dipoles in the propeller disk. His method later was extended and generalized by I. E. Garrick and C. E. Watkins (13, 1953) to the case of an in-flight propeller by considering the pressure dipoles that represent the thrust and torque force to be sub-

jected to a uniform rectilinear motion. The general theory of aerodynamic sound given by M. J. Lighthill (14, 1952 and 15, 1954) has been extended to situations containing arbitrarily moving boundaries in unbounded space by J. E. Ffowcs Williams and D. L. Hawkings (16, 1969), using the theory of generalized functions. The surface is replaced by a discontinuity in the flow-field, around which the motion of the fluid medium is assumed to be known. Other important works in rotational propeller noise are given in Refs. 17-25. In addition, K. Karamcheti and Y. H. Yu (26, 1974) have studied the hovering rotor propeller, minimizing the far field intensity subject to aerodynamic constraints.

This paper is concerned with the design of propellers for minimum noise. The paper is divided into three parts. In order to relate aerodynamic propeller design and propeller acoustics, the first part includes the necessary approximations and assumptions involved, the coordinate systems and their transformations, the geometry of the propeller blade, and the problem formulations including the induced velocity, required in the determination of mean lines of blade sections, and the optimization of propeller noise. The second part is devoted to the numerical formulation for the lifting-line model. The third part presents some applications and numerical results.

PART 1. BASIC THEORY AND FORMULATION

A. Assumptions and Consequences

An exact propeller design is not possible in any theoretical analysis so that a number of simplifying assumptions must be made. Except in certain special cases like stall-flutter, where nonlinearity is of essence, the aerodynamic tools should be mathematically linear to ensure the possibility of finding a solution with reasonable time and effort. The following assumptions, based on this concept, are generally made in theoretical propeller design and in the calculation of the sound pressure due to the moving propeller blades.

1. The propeller is operating in an unlimited stationary fluid with a constant advance velocity and a constant angular velocity.

2. The fluid is inviscid. Although all real fluids are compressible to a greater or lesser extent, under normal conditions the effects of compressibility are unimportant at low speed, and consequently the density of the fluid will be assumed to be constant in developing the vortex theory. However, from the acoustic viewpoint, the compressibility of fluid is important so that the fluid is restored to be compressible in the acoustic formulation.

3. The propeller consists of a set of identical, symmetrically spaced blades attached to a hub. The hub effect is ignored so that it is not necessary to satisfy the hub boundary conditions.

4. The blade sections are thin and the blade is not heavily loaded. In this case the disturbance velocities produced by the propeller are small compared with the propeller advance velocity and rotational velocity. Therefore, the deviation between the blade surfaces and the stream surfaces formed

by the relative undisturbed flow is small. This assumption permits us to treat the problem as a logical extension of linearized finite wing theory where the tangency condition is satisfied on the mean line of the profile. It also enables separation of the loading and thickness effects. However, from the acoustic viewpoint, this idea implies that the quadrupole sources, the Lighthill stress, are negligible, since they contain only those second-order perturbation terms which are dropped upon linearization.

5. Each propeller blade is replaced by a reference surface which is the projection of the actual blade outline on the helical surface with pitch angle β_i, the hydrodynamic advance angle obtained from the lifting line theory. A distribution of bound vortices for loading effects and sources and sinks for thickness effects are placed upon this reference surface. The vortices are distributed in both the chordwise and spanwise directions. The variation of strength of vorticity necessitates free vortices being shed from the bound vortices. These free vortices form helical surfaces behind the propeller and extend to infinity in the propeller-fixed coordinate system.

6. Upon neglecting the quadrupole sources, the blade loading and thickness are the only acoustic sources that will be considered.

7. The effects of slipstream contraction and centrifugal force on the shape of the free vortex sheets are ignored. Consequently, each of the free vortices has a constant diameter and constant pitch downstream which may be varied along the radius.

8. Body forces are ignored.

9. The two-dimensional chordwise pressure distributions are preserved in the three-dimensional flow.

We shall write all the quantities used in the formulation in non-dimensional form by referring all velocities to a reference velocity, V_p (V_p may be chosen to be the advance velocity of the propeller), and by referring all linear dimensions to a characteristic length, R_p, (propeller radius). The pressure and the force per unit area are made non-dimensional with respect to $\rho_o \; V_p^{\,2}$, where ρ_o is the undisturbed density of the fluid and time is referred to $1/\Omega$, where Ω is the angular velocity of the propeller. Also, the circulation is non-dimensionalized with respect to $2\pi R_p V_p$, and the strengths of the vortex sheets are referred to V_p. Further, the expression

$$\lambda_p = \frac{V_p}{R_p \Omega} \tag{1}$$

is referred to as the reference advance coefficient, which is the advance coefficient of the propeller if V_p is chosen to be the advance velocity of the propeller. Dimensional values are denoted by primes, so that, for example

$$t' = \frac{t}{\Omega} \tag{2}$$

B. Coordinate Systems

Two main coordinate systems, shown in Figs. 1 and 2 , are used in the analysis. One is the "space-fixed" coordinate system, which has a rectangular (x_1, x_2, x_3) coordinate system (x-system), a rectangular (y_1, y_2, y_3) coordinate system (y-system), and a spherical (ξ, Θ, Φ) coordinate system (ξ -system). The other one is the "propeller-fixed" coordinate system,

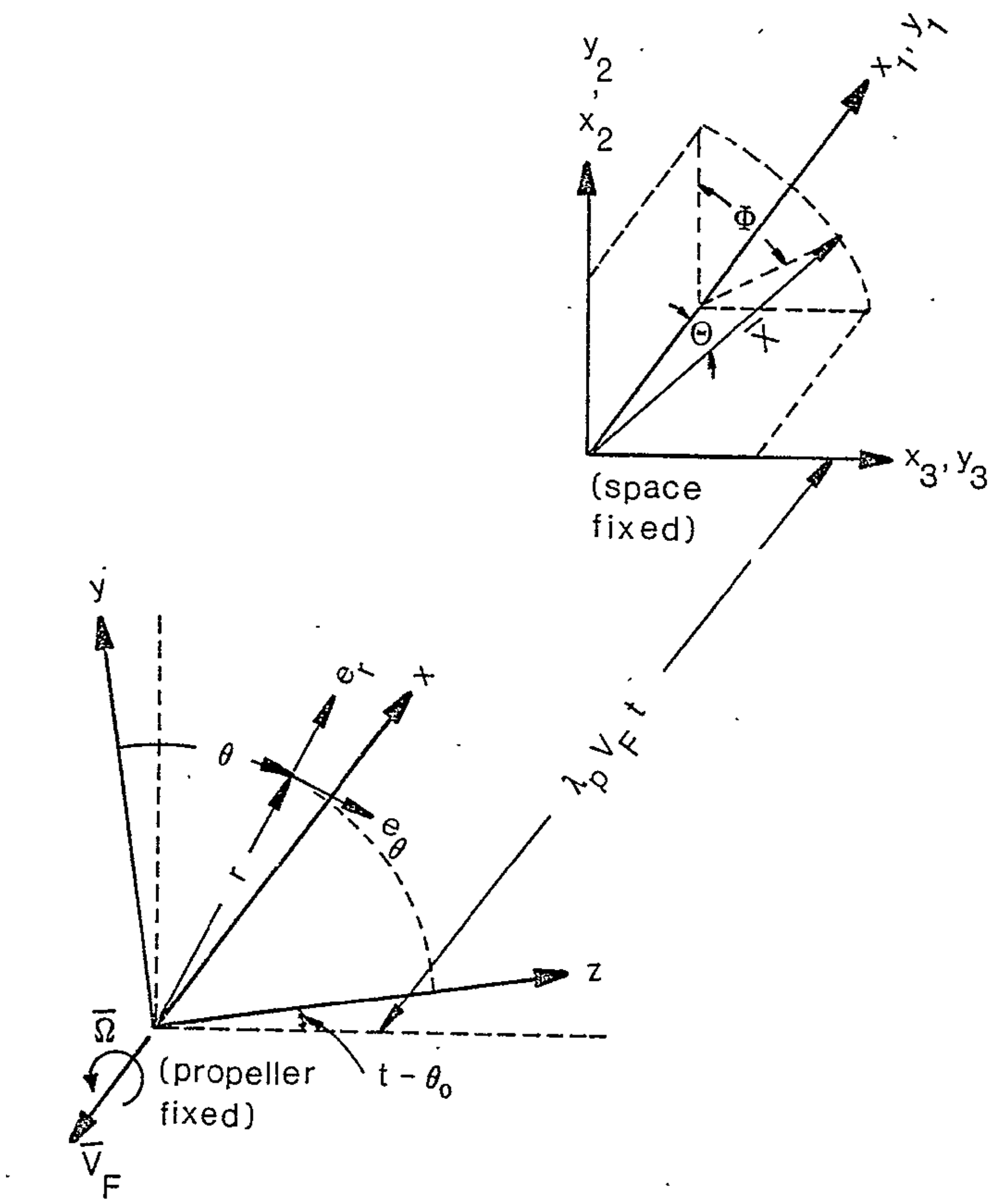

Fig. 1 Coordinate systems.

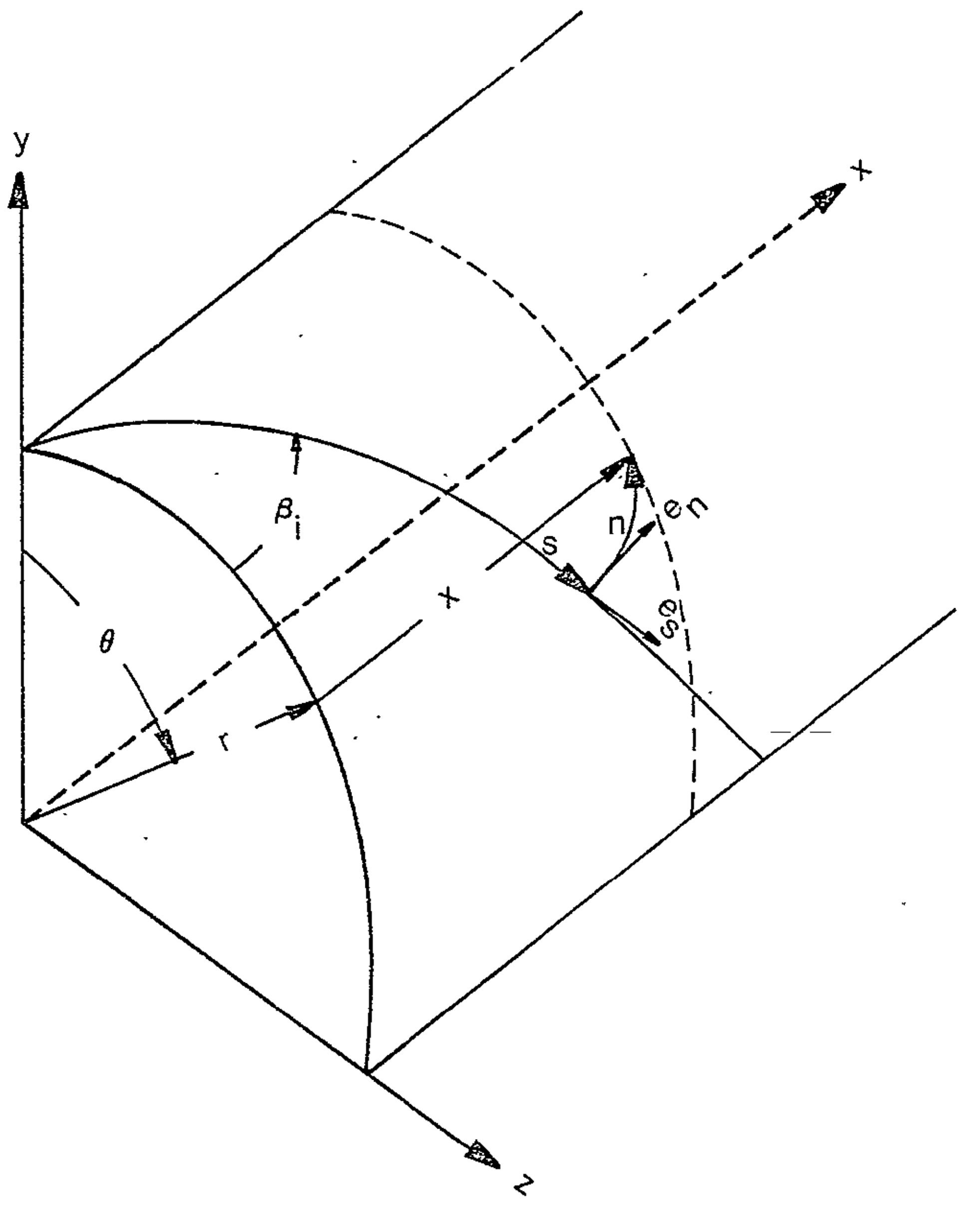

Fig. 2 Orthogonal curvilinear coordinate systems.

which has a rectangular (x, y, z) coordinate system (z-system), a cylindrical (x, r, θ) coordinate system (r-system), and a curvilinear (s, n, r) coordinate system (s-system). Also defined are the unit tangent vectors, e_w, to the coordinate curves w, where w = x_i, y_i, r, , ..., and so forth.

Propeller-Fixed Coordinate System:

All the propeller-fixed coordinates are attached to the propeller, translating and rotating with the propeller.

1. A rectangular (x, y, z) coordinate system (z-system)

   x-axis = axis of revolution of propeller with positive distance measured downstream

   y-axis = selected so as to pass through the tip of one blade

   z-axis = selected so as to complete the right-handed system

   $\overline{Z}$ = position vector of a space point referred to the center of the z-system

2. A cylindrical (x, r, θ) coordinate system (r-system)

   x-axis = defined as before

   r-coordinate = radial coordinate

   θ-coordinate = angular coordinate, measured clockwise starting from the y-axis when looking downstream

3. An orthogonal curvilinear (s, n, r) coordinate system (s-system)

   r-coordinate = defined as before

   s-coordinate = a helix whose non-dimensional pitch is

$$P_i = 2\pi r \tan\beta_i(r) = 2\pi\lambda_i(r) \tag{3}$$

where $\beta_i(r)$ is the hydrodynamic advance angle,

and $\lambda_i(r)$ is the hydrodynamic advance coefficient.

Furthermore, the s-coordinate is the intersection of the reference surface representing the first blade with a circular cylinder of radius r, measured downstream along the helix

n-coordinate = selected so as to complete the right-handed system

Space-Fixed Coordinate System:

1. A rectangular (x_1, x_2, x_3) coordinate system, referred to as an observer system (x-system)

   x_1-axis = axis of revolution of the propeller with positive distance measured downstream

   The x_1-axis, x_2-axis, and x_3-axis are fixed in space and complete a right-handed system. At time $t=0$, the origin of the x-system coincides with that of the z-system and the x_2-axis and y-axis make an angle θ_0 measured clockwise from the x_2-axis when looking downstream.

   $\overline{X}$ = position vector of an observer referred to the x-system

2. A rectangular (y_1, y_2, y_3) coordinate system, referred to as a source system (y-system)

   The y_1-axis, the y_2-axis, and the y_3-axis are selected so as to coincide with the x_1-axis, the x_2-axis, and the x_3-axis, respectively.

   $\overline{Y}$ = position vector of an acoustic source referred to

the y-system

3. A spherical (ξ, Θ, Φ) coordinate system, also referred to as an observer system (ξ-system)

ξ -coordinate = distance measured from the origin of the x-system to the observer

Φ -coordinate = angular coordinate, measured clockwise starting from x_2-axis when looking downstream

Θ -coordinate = angular coordinate, measured from the x_1-axis

$\bar{\xi}$ = $\bar{X}$

C. Some Useful Coordinate Transformations

Following are some useful coordinate transformations which will be used in the formulation of the problem. Also given are some relations among the unit tangent vectors of the different coordinate systems:

1. x-system and ξ-system

$$
\begin{aligned}
x_1 &= \xi \cos \Theta \\
x_2 &= \xi \cos \Phi \sin \Theta \\
x_3 &= \xi \sin \Phi \sin \Theta
\end{aligned}
\tag{4}
$$

2. z-system and y-system

$$
\begin{aligned}
y_1 &= x - \lambda_p V_F t \\
y_2 &= y \cos(t-\theta_0) + z \sin(t-\theta_0) \\
y_3 &= -y \sin(t-\theta_0) + z \cos(t-\theta_0)
\end{aligned}
\tag{5}
$$

$$
\begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t-\theta_0) & -\sin(t-\theta_0) \\ 0 & \sin(t-\theta_0) & \cos(t-\theta_0) \end{pmatrix} \begin{pmatrix} e_{y_1} \\ e_{y_2} \\ e_{y_3} \end{pmatrix}
\tag{6}
$$

3. z-system and r-system

$$
\begin{aligned}
x &= x \\
y &= r\cos\theta \\
z &= r\sin\theta
\end{aligned} \tag{7}
$$

$$
\begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} e_x \\ e_r \\ e_\theta \end{pmatrix}
$$

$$
\begin{pmatrix} e_x \\ e_r \\ e_\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} \tag{8}
$$

4. y-system and r-system

$$
\begin{aligned}
y_1 &= x - \lambda_p V_F t \\
y_2 &= r\cos(\theta + \theta_0 - t) \\
y_3 &= r\sin(\theta + \theta_0 - t)
\end{aligned} \tag{9}
$$

$$
\begin{pmatrix} e_x \\ e_r \\ e_\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta+\theta_0-t) & \sin(\theta+\theta_0-t) \\ 0 & -\sin(\theta+\theta_0-t) & \cos(\theta+\theta_0-t) \end{pmatrix} \begin{pmatrix} e_{y_1} \\ e_{y_2} \\ e_{y_3} \end{pmatrix} \tag{10}
$$

5. r-system and s-system

$$r = r$$

$$s = \frac{\lambda_i(r)x + r^2(\theta - \delta_k)}{\sqrt{r^2 + \lambda_i^2(r)}} \tag{11}$$

$$n = \frac{xr - r\lambda_i(r)(\theta - \delta_k)}{\sqrt{r^2 + \lambda_i^2(r)}}$$

where $\delta_k = \theta$ -coordinate of the point at the tip of the kth blade. For a symmetrical blade arrangement, these angles are

$$\delta_k = \frac{2\pi}{B}(k-1) \qquad k = 1, 2, \ldots, B \tag{12}$$

D. Geometrical Considerations

Figure 3 shows a projected view of the propeller, looking in the downstream direction (positive x). The angular coordinates θ_L and θ_T of the leading and trailing edges, respectively, define the projected blade outline.

The reference surface representing the kth blade is the projection of the actual blade outline on the helical surface:

$$H_k(x, r, \theta) = x - \lambda_i(r)(\theta - \delta_k) = 0 \tag{13}$$

An expanded view of the s-n plane showing a typical blade section oriented approximately along the s-coordinate curve is illustrated in Fig. 4.

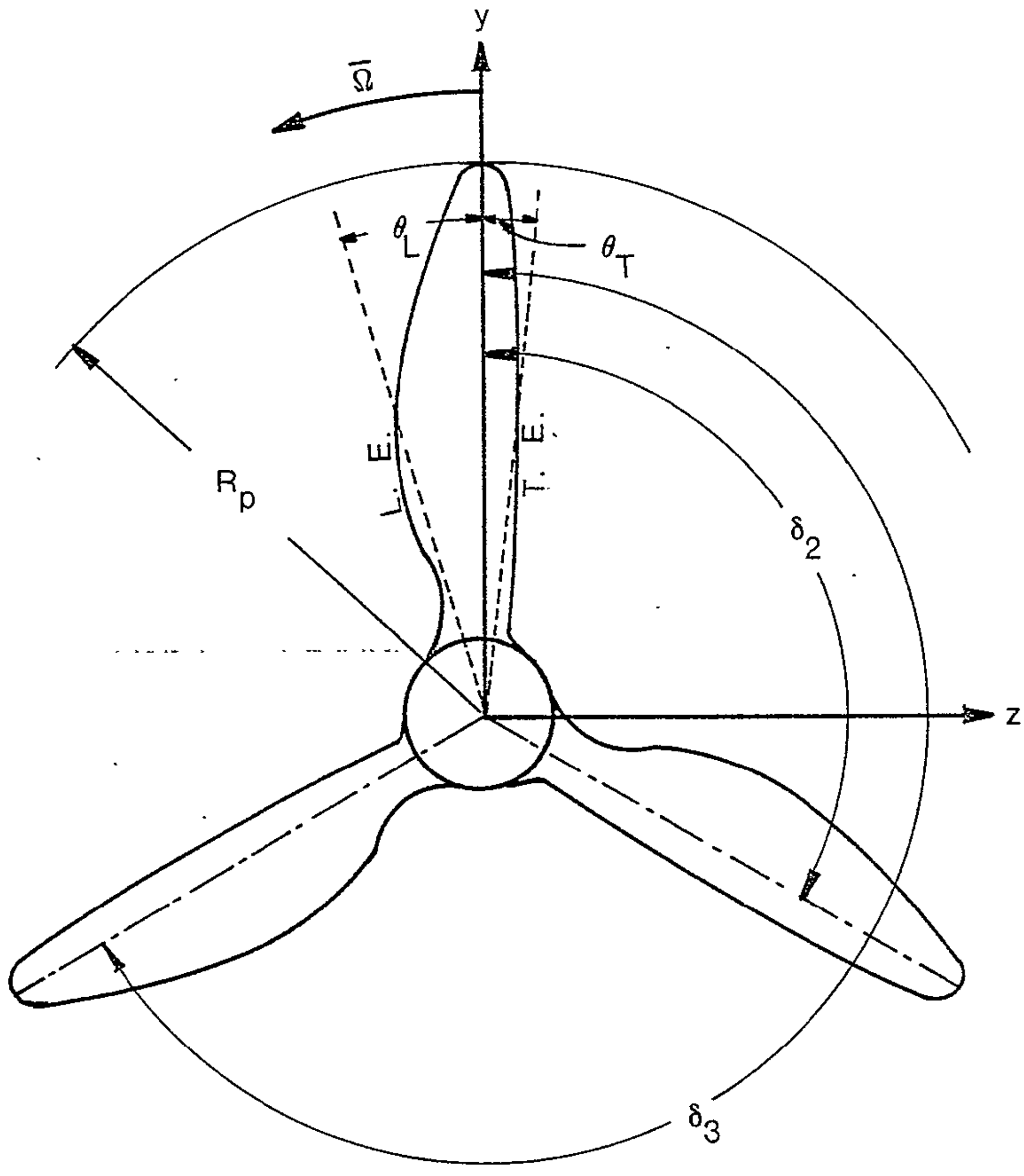

Fig. 3 Projected blade outlines; 3-bladed propeller.

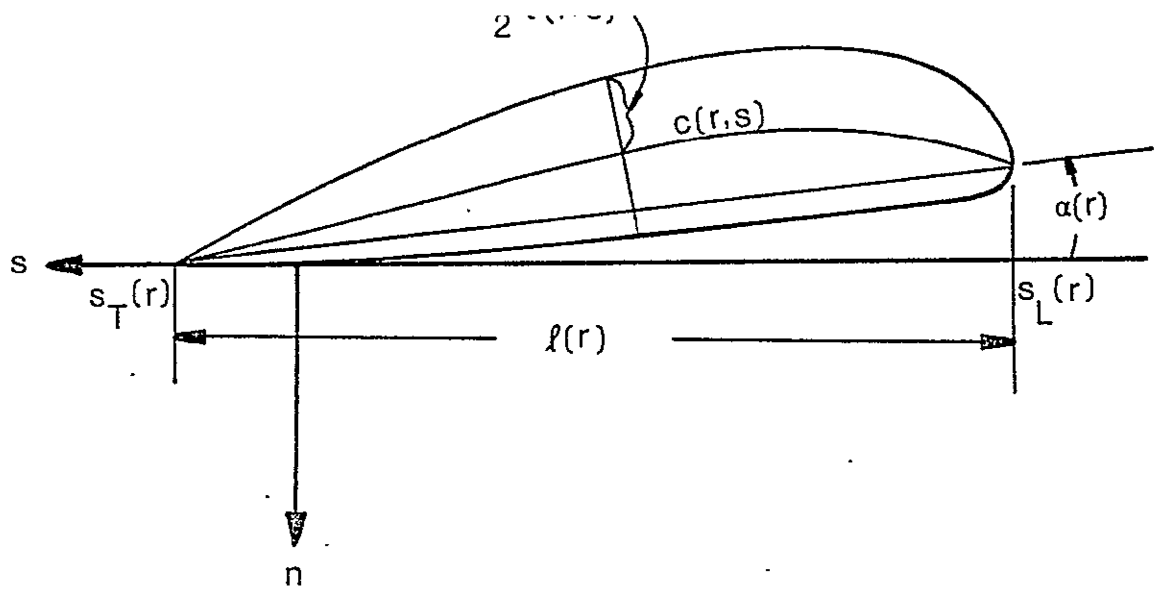

Fig. 4 Illustration of blade section.

Here $s_L(r)$ and $s_T(r)$ are the s-coordinates of the leading and trailing edges, while the total expanded chord length is $l(r)$. The incidence $\alpha(r)$ of the blade section at radius r is defined as the angle between the chord line of the section and the s-axis. The incidence is considered to be positive when the chord line has greater pitch than the reference surface. The mean line $c(r,s)$ and the thickness $t(r,s)$, are also shown. It is noted that the positioning of the actual section in the n-direction is immaterial since the blade section will be represented by singularities distributed along the s-axis.

The pitch of each of the reference surfaces is illustrated in Fig. 5. For a lightly loaded propeller, the strictly linearized case, these reference surfaces will coincide with those swept out by the undisturbed relative flow past the radial lines

$$\begin{aligned} \theta &= \delta_K \\ \chi &= 0 \end{aligned} \tag{14}$$

through the tip of each blade.

It is seen from Fig. 5 that the non-dimensional pitch is

$$P_o(r) = \frac{2\pi V_F V_P}{R_P \Omega} = 2\pi r \tan\beta(r) = 2\pi \lambda_F \tag{15}$$

where β = advance angle of the propeller

λ_F = advance coefficient of the propeller

Therefore, for the lightly loaded propeller theory, it is sufficient to set $P_i(r) = P_o(r)$. For the moderately loaded propeller, the nonlinear problem being approximated by an equivalent linear one with the consideration of perturbation velocities (induced velocities), the pitch of the

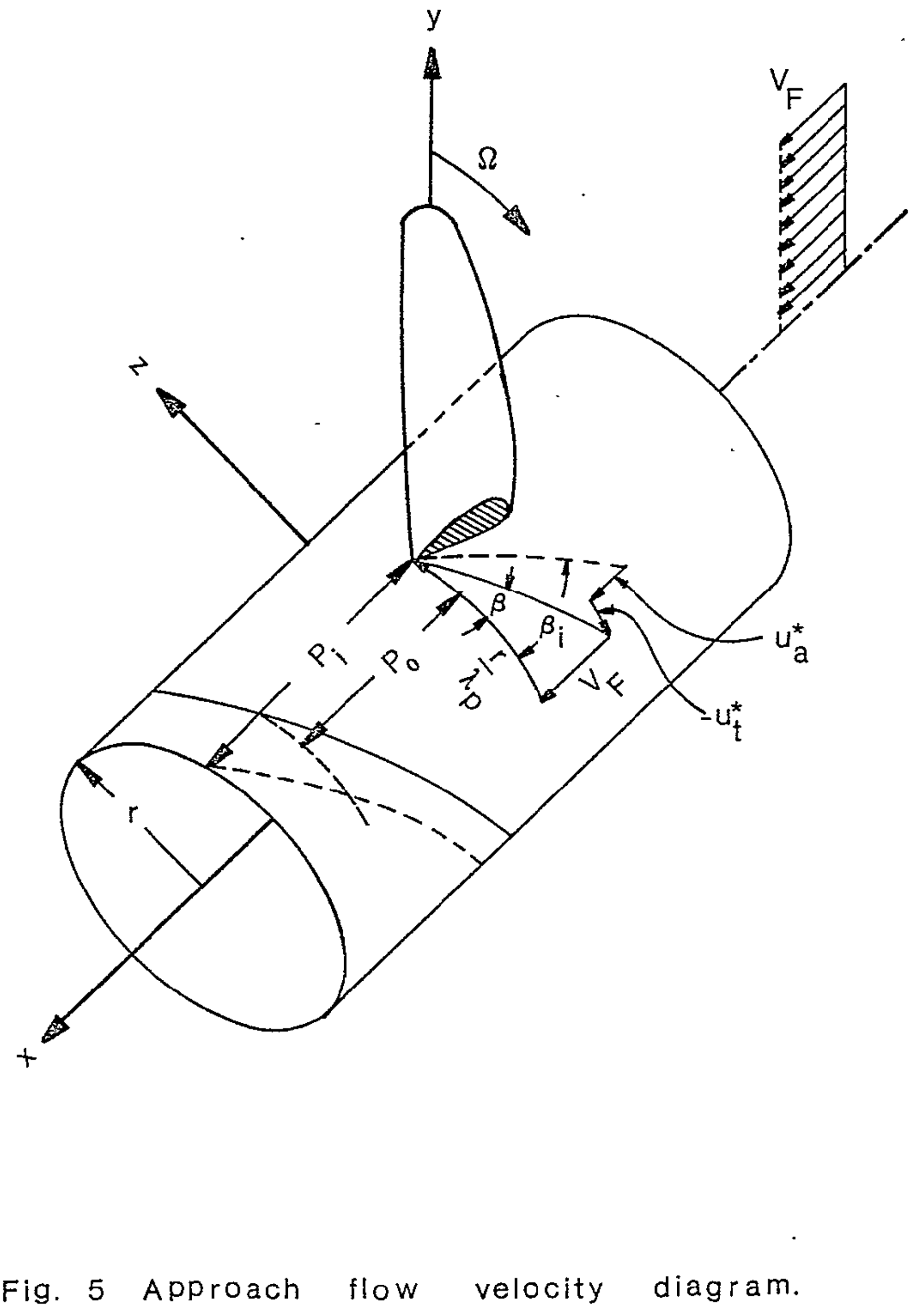

Fig. 5 Approach flow velocity diagram.

reference surface is increased over the lightly loaded case. From Fig. 5 it is seen that

$$P_i(r) = \frac{2\pi V_P(V_F + u_a^*(r))}{R_P\Omega + V_P\frac{u_t^*(r)}{r}} = \frac{2\pi r(V_F + u_a^*(r))}{\frac{r}{\lambda_P} + u_t^*(r)}$$

$$= 2\pi r \tan\beta_i(r) \tag{16}$$

$$= 2\pi\lambda_i(r)$$

where $\lambda_i(r)$ = hydrodynamic advance coefficient

$\beta_i(r)$ = hydrodynamic advance angle

$u_a^*(r)$ = axial component of induced velocity from lifting-line theory

$u_t^*(r)$ = tangential component of induced velocity from lifting line theory

The effective inflow velocity is

$$V^*(r) = \frac{V_F + u_a^*(r)}{\sin\beta_i(r)} = \frac{\frac{r}{\lambda_P} + u_t^*(r)}{\cos\beta_i(r)} \tag{17}$$

For convenience in the analysis which follows, information about the reference surface is given:

1. $(\lambda_i(\rho)\phi, \rho, \phi + \delta_k)$ = coordinate of any point on the kth reference surface, expressed in the r-system. ϕ is the angular coordinate of the corresponding point on the first reference surface.
2. The unit tangent vector at $(\lambda_i(\rho)\phi, \rho, \phi+\delta_k)$ has three components as follows:

$$e_r = \cos(\phi+\delta_K)\, e_y + \sin(\phi+\delta_K)\, e_z$$

$$e_s = \frac{1}{\sqrt{\rho^2+\lambda_i^2(\rho)}} \left\{ \lambda_i(\rho)\, e_x - \rho \sin(\phi+\delta_K)\, e_y + \rho \cos(\phi+\delta_K)\, e_z \right\}$$

$$e_n = \frac{1}{\sqrt{\rho^2+\lambda_i^2(\rho)}} \left\{ \rho\, e_x + \lambda_i(\rho) \sin(\phi+\delta_K)\, e_y - \lambda_i(\rho) \cos(\phi+\delta_K)\, e_z \right\} \tag{18}$$

3. $dA = \sqrt{\rho^2 + \lambda_i^2(\rho)}\; d\phi\, d\rho$ (19)

= infinitesimal area element of the reference surface at $(\lambda_i(\rho)\phi,\ \rho,\ \phi+\delta_K)$

4. $ds = \sqrt{\rho^2 + \lambda_i^2(\rho)}\; d\phi$ (20)

= infinitesimal line element along the helix at $(\lambda_i(\rho)\phi,\ \rho,\ \phi+\delta_K)$

5.

$$\sin\beta_i(\rho) = \frac{V_F + u_a^*(\rho)}{V^*(\rho)} = \frac{\lambda_i(\rho)}{\sqrt{\rho^2+\lambda_i^2(\rho)}}$$

$$\cos\beta_i(\rho) = \frac{\frac{\rho}{\lambda_P} + u_t^*(\rho)}{V^*(\rho)} = \frac{\rho}{\sqrt{\rho^2+\lambda_i^2(\rho)}} \tag{21}$$

It is noticed that ρ, ϕ are the dummy cylindrical coordinates of the point on the first reference surface.

E. Mathematical Formulation

This section is concerned with the formulation of the problem. It is divided into three subsections. The first one will be referred to as the aerodynamic formulation, dealing with the calculation of the mean lines of the blade sections. The second one will be referred to as the acoustic formulation, dealing with the acoustic problem caused by the moving propeller. The third subsection is concerned with formulation of

the criteria for optimization of propeller noise.

E-1. Aerodynamic Formulation

The mean line of the blade section relative to the helical chord at radius r is determined by the relative induced velocity normal to the reference surface $(\overline{V}^{(bt)}(r,\phi) - \overline{V}_L^*(r))$, where $\overline{V}^{(bt)}(r,\phi)$ is the total induced velocity obtained from lifting-surface theory and $\overline{V}_L^*(r)$ is the induced velocity obtained from lifting-line theory. We shall formulate this problem following closely the work of Kerwin and Leopold (10, 1964), based on the assumptions made in Section A.

Lifting Surface Theory:

1. Vortex Distribution

The total bound circulation around the blade section at radius r will be defined as $\Gamma(r)$ so that

$$L(r) = \int_{s_L(r)}^{s_T(r)} \Delta p(r,s)\, ds = V^*(r)\int_{s_L(r)}^{s_T(r)} \gamma(r,s)\, ds = 2\pi V^*(r)\,\Gamma(r) \tag{22}$$

where $L(r)$ = non-dimensional total lift force per unit radius, $L'(r)/\rho_o V_p^2 R_p$

$\Delta p(r,s)$ = non-dimensional pressure differential due to velocity discontinuity, $\Delta p'/\rho_o V_p^2$

$\gamma(r,s)$ = non-dimensional strength of the radially oriented bound vortices that induce a discontinuity in the streamwise velocity of $\pm$ 1/2 $\gamma(r,s)$ at each point on the reference surface, $\gamma'(r,s)/V_p$.

$\Gamma(r)$ = non-dimensional circulation, $\Gamma'(r)/2\pi R_p V_p$.

To satisfy the Helmholtz law of continuity of vortices, this system of bound vortices must be accompanied by a system of trailing vortices whose axis is in the e_s direction along a helix of pitch $P_i(r)$.

If the strength of the helical vortex sheet is defined as $\gamma_s(r)$ behind the trailing edge, then

$$\gamma_s(r) = -2\pi \frac{d\Gamma(r)}{dr} \qquad \theta_T \leq \theta \tag{23}$$

With this expression and Eq. (22), the strengths of the free vortices shed from the blade are found to be

$$\gamma_s(r) = -\frac{d}{dr}\int_{s_L(r)}^{s_T(r)} \gamma(r,s)\,ds = -\frac{d}{dr}\int_{\theta_L(r)}^{\theta_T(r)} \gamma(r,\theta)\sqrt{r^2+\lambda_i^2(r)}\,d\theta$$

$$= -\int_{\theta_L(r)}^{\theta_T(r)} \frac{\partial}{\partial r}\left\{\gamma(r,\theta)\sqrt{r^2+\lambda_i^2(r)}\right\} d\theta + \gamma(r,\theta_L)\sqrt{r^2+\lambda_i^2(r)}\,\frac{d\theta_L}{dr} \tag{24}$$

$$- \gamma(r,\theta_T)\sqrt{r^2+\lambda_i^2(r)}\,\frac{d\theta_T}{dr}$$

The first term on the right is due to the radial change in the bound vortices. The second term is due to the change of $\theta_L(r)$ with respect to r along the leading edge. It follows from this equation that, within the reference surface, the free vortex strength $\gamma_s(r, \theta)$ can be expressed as follows:

$$\gamma_s(r,\theta) = -\int_{\theta_L}^{\theta} \frac{\partial}{\partial r}\left\{\gamma(r,\phi)\sqrt{r^2+\lambda_i^2(r)}\right\} d\phi$$

$$+ \gamma(r,\theta_L)\sqrt{r^2+\lambda_i^2(r)}\,\frac{d\theta_L}{dr} \qquad \theta_L \leq \theta < \theta_T \tag{25}$$

It is evident that

$$\gamma_s(r,\theta) = -2\pi \frac{d\Gamma(r)}{dr} \qquad \theta_T \le \theta \tag{26}$$

2. Source Distribution

The thickness of the blades can be represented by a source sheet distribution. The connection between the source strength $\sigma(r,s)$, which induces a discontinuity in normal velocity $\pm\sigma(r,s)$ at any point on the surface, the effective inflow velocity $V^*(r)$ and the local change of the thickness with chordwise dimension is

$$\sigma(r,s) = V^*(r)\frac{\partial t(r,s)}{\partial s} \tag{27}$$

where $t(r,s)$ = non-dimensional thickness as shown in Fig. 6

$\sigma(r,s)$ = non-dimensional source strength, $\sigma'(r,s)/V_p$

3. Induced Velocities

As mentioned before, it is the main purpose of this section to compute the induced velocities due to loading and thickness. The computation of the induced velocities at points on the reference surfaces representing the blades of the propeller enables us to determine the way in which the blade sections should be cambered and oriented with respect to the effective inflow if a propeller is designed with a prescribed pressure loading and thickness.

3.1 Loading

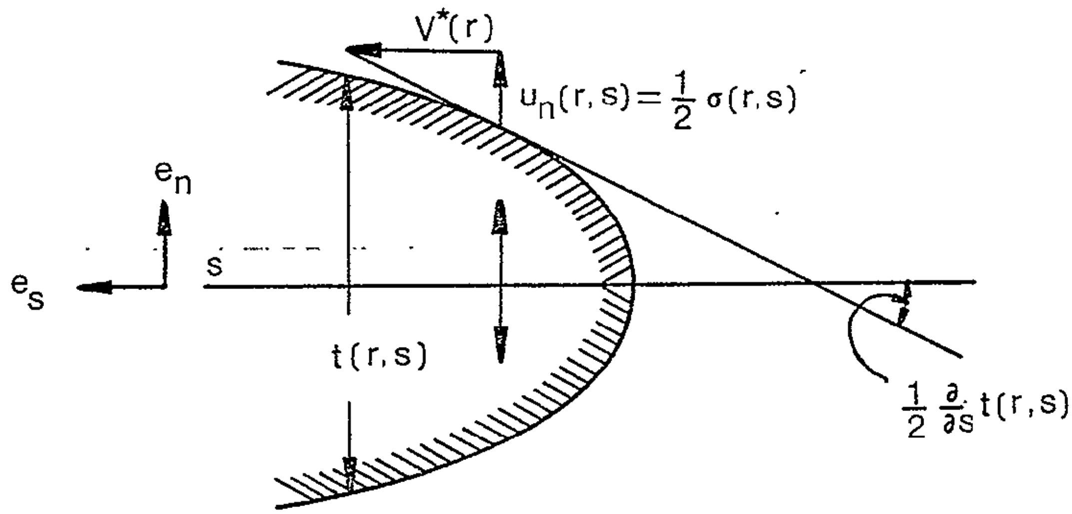

Fig. 6 Generation of a thickness form by sources.

From the Biot-Savart law, the induced velocities $\overline{V}^{(b)}$ and $\overline{V}^{(t)}$ at any point $P(\lambda_i(r)\theta, r, \theta)$ on the first blade by the bound vortices and trailing vortices, respectively, are found to be

$$\overline{V}^{(b)}(P) = \int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma(\rho,\phi)\sum_{K=1}^{B} \frac{e_r \times \overline{D}}{4\pi D^3}\, dA \tag{28}$$

$$\overline{V}^{(t)}(P) = \int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\infty} \gamma_s(\rho,\phi)\sum_{K=1}^{B} \frac{e_s \times \overline{D}}{4\pi D^3}\, dA \tag{29}$$

$$\overline{V}^{(bt)}(P) = \overline{V}^{(b)}(P) + \overline{V}^{(t)}(P) \tag{30}$$

where r_h = hub radius

$\overline{D} = \{\lambda_i(r)\theta - \lambda_i(\rho)\phi\} e_x + \{r\cos\theta - \rho\cos(\phi+\delta_K)\} e_y + \{r\sin\theta - \rho\sin(\phi+\delta_K)\} e_z$

the vector distance from source point $(\lambda_i(\rho)\phi, \rho, \phi+\delta_K)$ to the reference point $P(\lambda_i(r)\theta, r, \theta)$ on the reference surface.

$D = |\overline{D}|$

$\overline{V}^{(b)}(P)$ = induced velocity due to bound vortices

$\overline{V}^{(t)}(P)$ = induced velocity due to trailing vortices

$\overline{V}^{(bt)}(P)$ = total induced velocity due to loading

Upon submitting Eqs.(18) and (19) into Eqs. (28) and (29), evaluating the vector product, and converting velocity components into cylindrical coordinates, we have

$$u_a^{(b)}(r,\theta) = -\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma(\rho,\phi)\sum_{K=1}^{B} \frac{r\sin(\phi+\delta_K-\theta)}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\, d\phi\, d\rho \quad (31)$$

$$u_t^{(b)}(r,\theta) = -\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma(\rho,\phi)\sum_{K=1}^{B} \frac{\{\lambda_i(r)\theta-\lambda_i(\rho)\phi\}\cos(\phi+\delta_K-\theta)}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\, d\phi\, d\rho \quad (32)$$

$$u_r^{(b)}(r,\theta) = -\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma(\rho,\phi)\sum_{K=1}^{B} \frac{\{\lambda_i(r)\theta-\lambda_i(\rho)\phi\}\sin(\phi+\delta_K-\theta)}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\, d\phi\, d\rho \quad (33)$$

$$u_a^{(t)}(r,\theta) = \frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\infty} \gamma_s(\rho,\phi)\sum_{K=1}^{B} \frac{\rho^2-\rho r\cos(\phi+\delta_K-\theta)}{D^3}\, d\phi\, d\rho \quad (34)$$

$$u_t^{(t)}(r,\theta) = \frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\infty} \gamma_s(\rho,\phi)\sum_{K=1}^{B} \frac{\lambda_i(\rho)\{r-\rho\cos(\phi+\delta_K-\theta)\}+\rho\sin(\phi+\delta_K-\theta)\{\lambda_i(r)\theta-\lambda_i(\rho)\phi\}}{D^3}\, d\phi\, d\rho \quad (35)$$

$$u_r^{(t)}(r,\theta) = \frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\infty} \gamma_s(\rho,\phi)\sum_{K=1}^{B} \frac{\lambda_i(\rho)\rho\sin(\phi+\delta_K-\theta)+\{\lambda_i(r)\theta-\lambda_i(\rho)\phi\}\rho\cos(\phi+\delta_K-\theta)}{D^3}\, d\phi\, d\rho \quad (36)$$

where $\quad D \quad = \left\{ \left(\lambda_i(r)\theta - \lambda_i(\rho)\phi\right)^2 + r^2 + \rho^2 - 2\ r\rho \cos(\phi + \delta_K - \theta) \right\}^{1/2}$ (37)

$u_a^{(b)}(r, \theta)$ = non-dimensional axial component of induced velocity due to bound vortices

$u_t^{(b)}(r, \theta)$ = non-dimensional tangential component of induced velocity due to bound vortices

$u_r^{(b)}(r, \theta)$ = non-dimensional radial component of induced velocity due to bound vortices

$u_a^{(t)}(r, \theta)$ = non-dimensional axial component of induced velocity due to trailing vortices

$u_t^{(t)}(r, \theta)$ = non-dimensional tangential component of induced velocity due to trailing vortices

$u_r^{(t)}(r, \theta)$ = non-dimensional radial component of induced velocity due to trailing vortices

It should be noticed that the expressions for the induced velocity due to trailing vortices are different from those presented by Kerwin by a factor $(\rho^2 + \lambda_i^2(\rho))^{1/2}$.

3.2 Thickness

The velocity induced at any point $P(\lambda_i(r)\theta, r, \theta)$ on the first reference surface by sources distributed over B blades is found by taking the gradient of the velocity potential of the sources

$$\overline{V}^{(S)}(P) = \mathrm{grad} \int_{\rho = r_h}^{1} \int_{\phi = \theta_L(\rho)}^{\theta_T(\rho)} \sigma(\rho, \phi) \sum_{K=1}^{B} \frac{-1}{4\pi D}\ dA \tag{38}$$

Upon substituting Eq. (19) into Eq. (38), taking the gradient, and converting to cylindrical coordinates, one has

$$u_a^{(s)}(r,\theta)=\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sigma(\rho,\phi)\sum_{K=1}^{B}\frac{\lambda_i(r)\theta-\lambda_i(\rho)\phi}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\,d\rho \tag{39}$$

$$u_t^{(s)}(r,\theta)=\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sigma(\rho,\phi)\sum_{K=1}^{B}\frac{-\rho\sin(\phi+\delta_K-\theta)}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\,d\rho \tag{40}$$

$$u_r^{(s)}(r,\theta)=\frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sigma(\rho,\phi)\sum_{K=1}^{B}\frac{r-\rho\cos(\phi+\delta_K-\theta)}{D^3}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\,d\rho \tag{41}$$

where

$u_a^{(s)}(r,\theta)$ = non-dimensional axial component of induced velocity due to thickness

$u_t^{(s)}(r,\theta)$ = non-dimensional tangential component of induced velocity due to thickness

$u_r^{(s)}(r,\theta)$ = non-dimensional radial component of induced velocity due to thickness

3.3 Induced Velocity Normal to Blade

As mentioned before, in order to calculate the mean lines of the blade sections, the normal component, u_n, of the total induced velocity due to loading and thickness is required at any point $P(\lambda_i(r)\theta,\ r,\ \theta)$ on the first blade. This normal component of the induced velocity is related to the axial and tangential components of the total induced velocity at that

point by the expression:

$$u_n(r,\theta) = \left(\bar{V}^{(bt)}(r,\theta) + \bar{V}^{(s)}(r,\theta)\right) \cdot e_n$$

$$= u_n^{(b)}(r,\theta) + u_n^{(t)}(r,\theta) + u_n^{(s)}(r,\theta) \tag{42}$$

$$u_n^{(b)}(r,\theta) = \bar{V}^{(b)}(r,\theta) \cdot e_n = \frac{r\,u_a^{(b)}(r,\theta) - \lambda_i(r)\,u_t^{(b)}(r,\theta)}{\sqrt{r^2 + \lambda_i^2(r)}} \tag{43}$$

$$u_n^{(t)}(r,\theta) = \bar{V}^{(t)}(r,\theta) \cdot e_n = \frac{r\,u_a^{(t)}(r,\theta) - \lambda_i(r)\,u_t^{(t)}(r,\theta)}{\sqrt{r^2 + \lambda_i^2(r)}} \tag{44}$$

$$u_n^{(s)}(r,\theta) = \bar{V}^{(s)}(r,\theta) \cdot e_n = \frac{r\,u_a^{(s)}(r,\theta) - \lambda_i(r)\,u_t^{(s)}(r,\theta)}{\sqrt{r^2 + \lambda_i^2(r)}} \tag{45}$$

4. Mean Line Calculation (Boundary Conditions)

The mean line of the blade section at radius r can be obtained by considering the boundary conditions at the blade surface. The boundary condition is that the resultant velocity at any point on the blade surface must be tangent to the surface at that point. In the case of linearized theory, the surface to which the resultant velocity is tangent, at a given radius and chordwise position, is defined to be the surface tangent

to the mean line at that point as shown in Fig. 7.

From Fig. 7 and within the concept of linearized theory, the boundary condition at any point on the blade surface is

$$\frac{\partial}{\partial s} c_m(r,s) = \frac{u_n(r,s) - u_n^*(r)}{V^*(r)} \tag{46}$$

where

$c_m(r,s)$ = ordinate of the mean line of the blade section at radius r relative to the chord, measured in the e_n direction starting from the helical chord.

$u_n^*(r)$ = normal component of induced velocity from lifting-line theory

Upon integrating Eq. (46), we have

$$c_m(r,s) = -\int_{s}^{s_T(r)} \frac{u_n(r,s) - u_n^*(r)}{V^*(r)}\, ds \tag{47}$$

$$\frac{c_m(r,s)}{\sqrt{r^2+\lambda_i^2(r)}} = -\int_{\phi=\theta}^{\theta_T(r)} \frac{u_n(r,\phi) - u_n^*(r)}{V^*(r)}\, d\phi \tag{48}$$

Introducing the camber c(r,s) and the ideal angle of attack $\alpha_i(r)$, Eq. (47) may be expressed as

$$c(r,s) + (s_T - s)\tan\alpha_i(r) = \int_{s}^{s_T(r)} \frac{u_n(r,s) - u_n^*(r)}{V^*(r)}\, ds \tag{49}$$

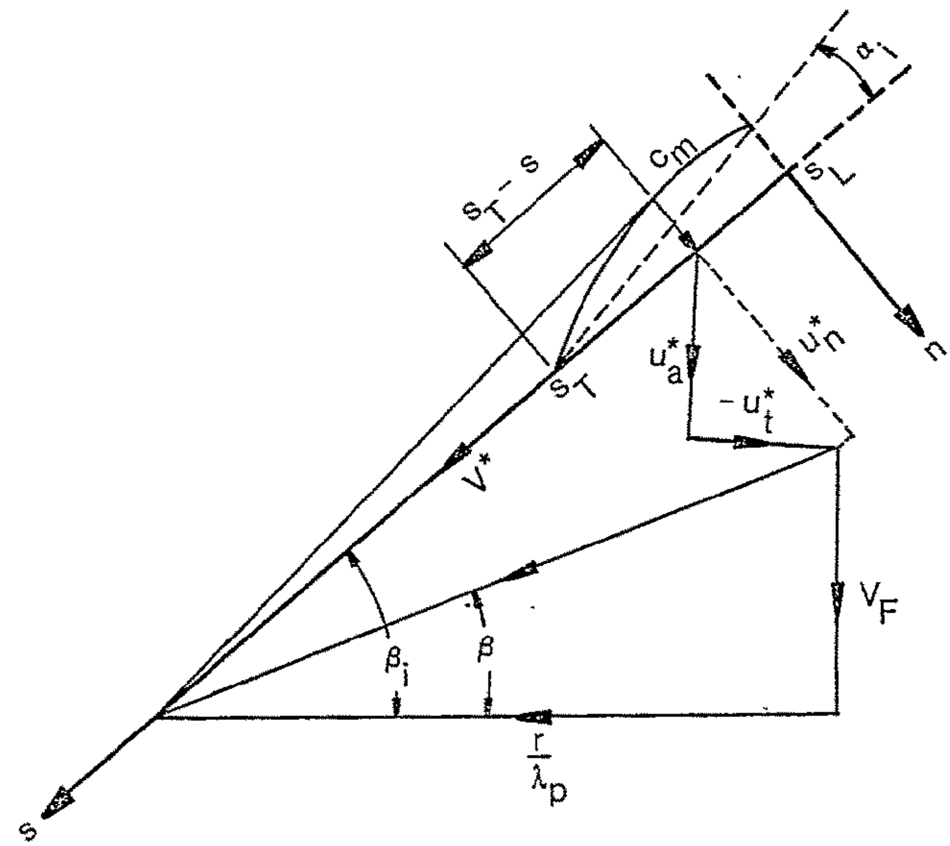

Fig. 7 Mean line and boundary condition.

To determine $\alpha_i(r)$, this integral is evaluated from the leading edge to the trailing edge and c(r,s) is taken to be zero. Having obtained the mean line, the blade section is obtained by adding the thickness to it.

Lifting-Line Theory:

As described in the preceding section, the total induced velocity normal to the blade surface can be divided into two parts. One is the velocity, $u_n^*(r)$, due to the lifting-line helical vortex sheet model. The other is the velocity resulting from spreading out the concentrated vortex lines to the desired blade outline and adding thickness to the blade. In order to obtain the second part, $u_n^*(r)$ must be obtained first. The assumptions underlying the theory of moderately loaded propellers permit one to design a propeller either to produce a given thrust or to absorb a given power, and to compute the ideal thrust and power and, therefore, efficiency, utilizing only the lifting-line representation of the propeller.

1. Lifting-Line Induced Velocity

As mentioned before, we are concerned with a hubless lifting-line propeller so that only the relationship between circulation and lifting line disturbance velocity for a hubless propeller is determined.

Since the lifting-line model of the propeller is a degenerate case of the lifting-surface model, the induced velocity components may be obtained from Eqs. (31) through (36). The induced velocity due to thickness is ignored. With the assumptions that the propeller consists of a set of identical, symmetrically spaced blades attached to a hub, the induced velocity components associated with thickness and chordwise distribution on the blade disappear, leaving only the wake term. Introducing the

lifting-line induction factors (8, 1952), we have the induced velocity components at the first lifting line

$$u_a^*(r) = -\frac{1}{2}\int_{\rho=r_h}^{1} \frac{d\Gamma(\rho)}{d\rho}\left(\frac{I_a(r,\rho)}{\rho-r}\right) d\rho \tag{50}$$

$$u_t^*(r) = -\frac{1}{2}\int_{\rho=r_h}^{1} \frac{d\Gamma(\rho)}{d\rho}\left(\frac{I_t(r,\rho)}{\rho-r}\right) d\rho \tag{51}$$

$$u_r^*(r) = -\frac{1}{2}\int_{\rho=r_h}^{1} \frac{d\Gamma(\rho)}{d\rho}\left(\frac{I_t(r,\rho)}{\rho-r}\right) d\rho \tag{52}$$

$$\begin{aligned} u_n^*(r) &= u_a^*(r)\cos\beta_i(r) - u_t^*(r)\sin\beta_i(r) \\ &= -\frac{1}{2}\int_{\rho=r_h}^{1} \frac{d\Gamma(\rho)}{d\rho}\left(\frac{I_n(r,\rho)}{\rho-r}\right) d\rho \end{aligned} \tag{53}$$

$$\overline{V}_L^*(r) = u_a^*(r)\, e_x + u_t^*(r)\, e_\theta + u_r^*(r)\, e_r$$

where

$u_a^*(r)$ = lifting-line axial component of induced velocity

$u_t^*(r)$ = lifting-line tangential component of induced velocity

$u_r^*(r)$ = lifting-line radial component of induced velocity

$u_n^*(r)$ = lifting-line normal component of induced velocity

$I_a(r,\rho)$ = lifting-line axial induction factor

$I_t(r,\rho)$ = lifting-line tangential induction factor

$I_r(r,\rho)$ = lifting-line radial induction factor

$I_n(r,\rho)$ = lifting-line normal induction factor

$⨍$ = Cauchy principal value integral

The lifting line induction factors are defined as

$$I_a(r,\rho) = ⨍_{\mu=0}^{\infty} \sum_{K=1}^{B} \frac{(\rho-r)\{\rho^2 - \rho r \cos(\delta_K+\mu)\}}{D^{*3}} \, d\mu \tag{54}$$

$$I_t(r,\rho) = ⨍_{\mu=0}^{\infty} \sum_{K=1}^{B} \frac{(\rho-r)\lambda_i(\rho)\{r - \rho\cos(\delta_K+\mu) - \rho\mu\sin(\delta_K+\mu)\}}{D^{*3}} \, d\mu \tag{55}$$

$$I_r(r,\rho) = ⨍_{\mu=0}^{\infty} \sum_{K=1}^{B} \frac{(\rho-r)\lambda_i(\rho)\{\rho\sin(\delta_K+\mu) - \rho\mu\cos(\delta_K+\mu)\}}{D^{*3}} \, d\mu \tag{56}$$

$$I_n(r,\rho) = \frac{1}{\sqrt{r^2+\lambda_i^2(r)}} \{ r\, I_a(r,\rho) - \lambda_i(r)\, I_t(r,\rho) \} \tag{57}$$

where

$$D^* = \{ \rho^2 + r^2 - 2\rho r \cos(\delta_K+\mu) + \mu^2 \lambda_i^2(\rho) \}^{1/2} \tag{58}$$

$$\lambda_i(r) = \frac{r\left\{V_F + u_a^*(r)\right\}}{\frac{r}{\lambda_p} + u_t^*(r)} \tag{59}$$

It is evident from Eq. (53) and the expressions for the induction factors that the components of induced velocity can not easily be obtained from Eq. (53) for a given circulation $\Gamma(r)$ since all induction factors are functions of $\lambda_i(r)$ which, in turn, depends on the axial and tangential components of the induced velocity through Eq. (59). In order to obtain these components of induced velocity, an iterative scheme, which will be discussed in the second part, must be applied. A brief summary of the evaluation of induction factors may be found in Appendix A.

2. Thrust and Power Coefficients

Shown in Fig. 8 are the velocity and force components at each radius according to the theory of moderately loaded propellers. The lift L(r) can be expressed in terms of bound circulation $\Gamma(r)$ (see Eq. 22) as

$$L(r) = 2\pi V^*(r)\,\Gamma(r) \tag{60}$$

The effective inflow velocity V*(r) can be written in terms of u_a^*, u_t^*, Ω, and $\beta_i(r)$ as follows

$$V^*(r) = \frac{\frac{r}{\lambda_p} + u_t^*(r)}{\cos\beta_i(r)} \tag{61}$$

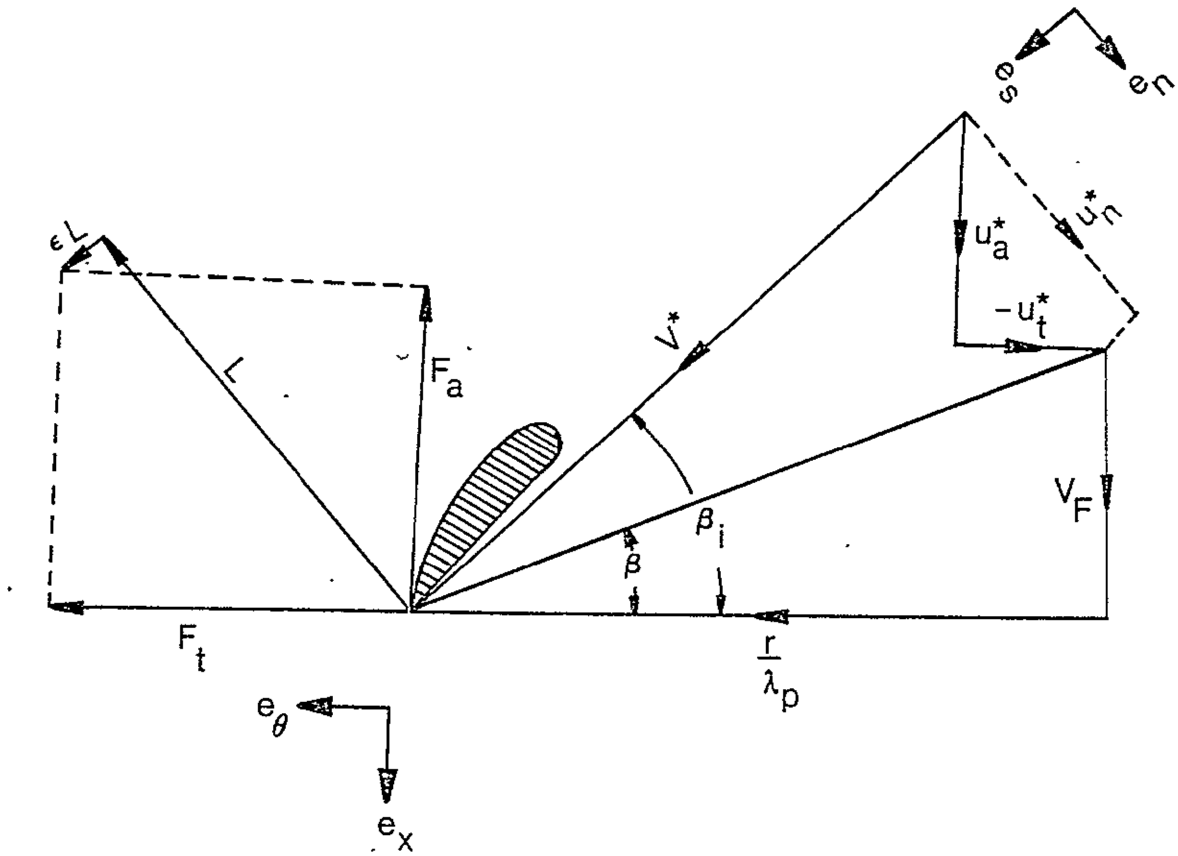

Fig. 8 Force and velocity diagram

or

$$V^*(r) = \frac{V_F + u_a^*(r)}{\sin\beta_i(r)} \tag{62}$$

From Fig. 8, we obtain

$$F_a(r) = L(r)\cos\beta_i(r)\,\{1 - \epsilon(r)\tan\beta_i(r)\} \tag{63}$$

$$F_t(r) = L(r)\sin\beta_i(r)\,\{1 + \frac{\epsilon(r)}{\tan\beta_i(r)}\} \tag{64}$$

where

$F_a(r)$ = axial component of force, per unit radius acting on the blade section at r

$F_t(r)$ = tangential component of force, per unit radius acting on the blade section at r

$\epsilon(r)$ = drag/lift ratio of the blade section at r

Upon substitution Eqs. (60), (61), and (62) into Eqs. (63) and (64), and integrating both Eqs. (63) and (64), we have the thrust and power coefficients, non-dimensionalized with respect to the reference velocity V_p and the radius of the propeller, as follows

$$C_T = \frac{T}{\frac{1}{2}\rho_0 V_p^2 \pi R_p^2} = 4B\int_{r=r_h}^{1}\Gamma(r)\left\{\frac{r}{\lambda_p} + u_t^*(r)\right\}\{1-\epsilon(r)\tan\beta_i(r)\}\,dr \tag{65}$$

$$C_P = \frac{P}{\frac{1}{2}\rho_o V_p^3 \pi R_p^2} = \frac{4B}{\lambda_p}\int_{r=r_h}^{1}\Gamma(r)\{V_F + u_a^*(r)\}\{1 + \frac{\epsilon(r)}{\tan\beta_i(r)}\}r\,dr \tag{66}$$

where

T = total propeller thrust

P = total power absorbed

The efficiency of the propeller is the ratio of power output to the power input which in this case is simply

$$\eta = \frac{C_T}{C_P} V_F \tag{67}$$

Finally, the ideal thrust coefficient, the ideal power coefficient, and the ideal efficiency are obtained directly from Eqs. (65),(66) and (67) by setting ϵ (r) equal to zero:

$$C_{Ti} = 4B\int_{r=r_h}^{1}\Gamma(r)\{\frac{r}{\lambda_p} + u_t^*(r)\}\,dr \tag{68}$$

$$C_{Pi} = \frac{4B}{\lambda_p}\int_{r=r_h}^{1}\Gamma(r)\{V_F + u_a^*(r)\}\,r\,dr \tag{69}$$

$$\eta_i = \frac{C_{Ti}}{C_{Pi}} V_F \tag{70}$$

Separation of Lifting-Line and Lifting Surface Velocities

The difficulty of evaluating the integral of Eq. (29) (or the integrals of (34) (35) and (36) to infinity may be avoided by separating the trailing vortex strength $\gamma_s(r,\theta)$ into lifting-line and lifting-surface parts (Refs. 9, 10, and 11). This is done by adding to and subtracting from Eq. (29) the quantity

$$\frac{1}{2}\int_{\rho=r_h}^{1}\int_{\phi=\theta}^{\theta_T}\frac{d\Gamma(\rho)}{d\rho}\sum_{k=1}^{B}\frac{e_s\times\bar{D}}{D^3}\,dA$$

The result may be expressed as

$$\bar{V}^{(t)}(r,\theta)=\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\gamma_s^{*}(\rho,\phi)\sum_{k=1}^{B}\frac{e_s\times\bar{D}}{4\pi D^3}\,dA+\bar{V}_L^{*}(r,\theta) \tag{71}$$

where

$$\begin{aligned}\gamma^{*}(\rho,\phi) &= \gamma_s(\rho,\phi) && \theta_L \leq\phi\leq\theta\\ &= \gamma_s(\rho,\phi)+2\pi\frac{d\Gamma(\rho)}{d\rho} && \theta<\phi\leq\theta_T\end{aligned}$$

$$\bar{V}_L^{*}(r,\theta)=\frac{-1}{2}\int_{\rho=r_h}^{1}\int_{\phi=\theta}^{\infty}\frac{d\Gamma(\rho)}{d\rho}\sum_{k=1}^{B}\frac{e_s\times\bar{D}}{D^3}\,dA \tag{72}$$

It can be easily shown that vector $\bar{V}_L^{*}(r,\theta)$ is equal to the induced velocity $\bar{V}_L^{*}(r)$ at radius r if the hydrodynamic advance coefficient $\lambda_i(r)$ is constant. For the case that $\lambda_i(r)$ is not constant, the approximation

will be made that $\overline{V}_L^*(r,\theta)$ can be replaced by $\overline{V}_L^*(r)$ and considered to be constant as the point P moves along a helix. With this approximation, Eq. (48) may be rewritten as

$$\frac{c_m(r,\theta)}{\sqrt{r^2+\lambda_i^2(r)}} = -\int_{\phi=\theta}^{\theta_T(r)} \frac{\bar{u}_n(r,\phi)}{V^*(r)}\, d\phi \tag{73}$$

where

$$\bar{u}_n(r,\theta) = u_n^{(b)}(r,\theta) + \bar{u}_n^{*(t)}(r,\theta) + u_n^{(s)}(r,\theta)$$

and

$$\begin{aligned}\bar{u}_n^{*(t)}(r,\theta) &= \left(\overline{V}^{(t)}(r,\theta) - \overline{V}_L^*(r,\theta)\right)\cdot e_n \\ &= \frac{r\,\bar{u}_a^*(r,\theta) - \lambda_i(r)\,\bar{u}_t^*(r,\theta)}{\sqrt{r^2+\lambda_i^2(r)}}\end{aligned} \tag{74}$$

$$\bar{u}_a^{*(t)}(r,\theta) = \frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma_s^*(\rho,\phi)\sum_{K=1}^{B}\frac{\rho^2-\rho r\cos(\phi+\delta_K-\theta)}{D^3}\, d\phi\, d\rho \tag{75}$$

$$\bar{u}_t^{*(t)}(r,\theta) = \frac{1}{4\pi}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} \gamma_s^*(\rho,\phi)\sum_{K=1}^{B}\frac{\lambda_i(\rho)\{r-\rho\cos(\phi+\delta_K-\theta)\}+\{\lambda_i(r)\theta-\lambda_i(\rho)\phi\}\rho\sin(\phi+\delta_K-\theta)}{D^3}\, d\rho\, d\phi$$

(76)

Substituting Eqs. (43), (45), and (74) into Eq. (73), we have

$$C_m(r,\theta) = -\frac{1}{V^*(r)} \int_{\theta}^{\theta_T(r)} \Big\{ r\big(u_a^{(b)}(r,\phi) + \bar{u}_a^{*(t)}(r,\phi) + u_a^{(s)}(r,\phi)\big)$$

$$- \lambda_i(r)\big(u_t^{(b)}(r,\phi) + \bar{u}_t^{*(t)}(r,\phi) + u_t^{(s)}(r,\phi)\big)\Big\}\, d\phi \tag{77}$$

The integration with respect to ϕ for evaluation of the integrand of Eq. (77) is now limited within the bounds of $\theta_L(r)$ and $\theta_T(r)$.

E-2. Acoustic Formulation

The starting point of the acoustic formulation for propeller noise is an exact Ffowcs Williams-Hawkings governing equation of motion, which is a formal statement of the generalized basic equations of motion, concerning the sound generated by turbulence and by a surface in arbitrary motion (Refs. 16 and 23). This integral representation is referred to as the FW-H equation (18, 1974). The FW-H equation shows that for a moving rigid body the acoustic density perturbation ($\rho' - \rho_o$) at a point $\bar{X}$ in the space-fixed coordinate system (x-system) and time t is given by

$$\frac{4\pi a_o^2(\rho' - \rho_o)}{\rho_o V_p^2} = \frac{\partial^2}{\partial x_i \partial x_j} \int_{\upsilon(t_o)} \left[\frac{T_{ij}}{R\left|1 - \frac{\bar{R}}{R}\cdot\bar{M}\right|} \right]_{\tau=\tau_e} d\upsilon(\bar{Z})$$

$$- \frac{\partial}{\partial x_i} \int_{S(t_o)} \left[\frac{f_i}{R\left|1 - \frac{\bar{R}}{R}\cdot\bar{M}\right|} \right]_{\tau=\tau_e} dA(\bar{Z})$$

$$- \frac{\partial}{\partial x_j} \int_{\upsilon_c(t_o)} \left[\frac{a_j/\lambda_p}{R\left|1 - \frac{\bar{R}}{R}\cdot\bar{M}\right|} \right]_{\tau=\tau_e} d\upsilon(\bar{Z})$$

$$+ \frac{\partial^2}{\partial x_i \partial x_j} \int_{\upsilon_c(t_o)} \left[\frac{V_i V_j}{R\left|1 - \frac{\bar{R}}{R}\cdot\bar{M}\right|} \right]_{\tau=\tau_e} d\upsilon(\bar{Z}) \qquad (78)$$

where

a_o = speed of sound in undisturbed medium

ρ' = instantaneous density

$\overline{X}$ = vector position of field point in the x-system

$\overline{Y}$ = vector position of acoustic source point in the y-system

t_o = initial time

t = non-dimensional observer time

$\overline{R}$ = vector distance between observer and source point

R = magnitude of $\overline{R}$

$\mathcal{V}(t_o)$ = whole space at initial time t_o

$S(t_o)$ = moving surface

$\mathcal{V}_c(t_o)$ = volume enclosed by $S(t_o)$

T_{ij} = non-dimensional Lighthill stress tensor referred to $\rho_o V_p^2$

V_i = the ith velocity component of the moving acoustic source in the y-system

$\overline{V}$ = $(V_i) = \overline{V}_o + R_p \Omega \times \overline{Z}/V_p$ (for moving rigid body)

$\overline{\Omega}$ = angular velocity of z-system

$\overline{Z}$ = vector position of acoustic source in the z-system

$\overline{V}_o$ = translational velocity of z-system

$\overline{M}$ = $V_p \overline{V}/a_o$

a_j = the jth component of non-dimensional acceleration of the moving acoustic source in the y-system, non-dimensionalized with respect to $V_p \Omega$

τ = dummy time variable

τ_e = non-dimensional source time (retarded time)

M_p = reference Mach number (tip Mach number of propeller),

f_j = the jth component of force vector exerted on fluid by moving surface

The first term of Eq. (78) shows that each moving element $d\mathcal{V}(\overline{Z})$ outside $S(\overline{Z})$ is equivalent to a moving quadrupole source of strength $T_{ij}\ d\mathcal{V}(\overline{Z})$. The second term shows that each element of surface area $dS(\overline{Z})$ is equivalent to a moving dipole of strength $-f_j dA(\overline{Z})$. The last two terms show that each moving volume element $d\mathcal{V}(\overline{Z})$, within S acts as if it emitted elementary waves which are the same as those emitted by a dipole source of strength $-a_j/\lambda_p\, d\mathcal{V}(\overline{Z})$ and a quadrupole of strength $V_i V_j\, d\mathcal{V}(\overline{Z})$ and represent the

sound generated by the volume displacement effects.

The consequence of the thin blade assumption is that the quadrupole sources T_{ij} in Eq. (78) are negligible and will be set to zero. The surface integrals may be replaced by a single-sided integral over the reference surface, and the new source strength is the sum of the corresponding upper surface and lower surface sources. Furthermore, the integrands in the last two terms are evaluated at the reference surface and assumed to be independent of the n-coordinate, because the upper and lower surfaces are close together.

Introducing the r-system into Eq. (78) and separating the loading and thickness effects, we have

a. Acoustic pressure due to loading (force noise)

$$4\pi(p-p_0)^{(b)}(\bar{X},t,\theta_0) = -\frac{\partial}{\partial x_i}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sum_{K=1}^{B}\left[\frac{\gamma(\rho,\phi)\,f_i}{R\left|1-\frac{\bar{R}}{R}\cdot\bar{M}\right|}\right]_{\tau=\tau_e}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\, d\rho \tag{79}$$

b. Acoustic pressure due to thickness (thickness noise)

$$\begin{aligned}4\pi(p-p_0)^{(s)}(\bar{X},t,\theta_0) = &-\frac{\partial}{\partial x_j}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sum_{K=1}^{B}\left[\frac{(a_j/\lambda_p)\,t(\rho,\phi)}{R\left|1-\frac{\bar{R}}{R}\cdot\bar{M}\right|}\right]_{\tau=\tau_e}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\, d\rho \\ &+\frac{\partial^2}{\partial x_i\partial x_j}\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)}\sum_{K=1}^{B}\left[\frac{V_i V_j\, t(\rho,\phi)}{R\left|1-\frac{\bar{R}}{R}\cdot\bar{M}\right|}\right]_{\tau=\tau_e}\sqrt{\rho^2+\lambda_i^2(\rho)}\;d\phi\, d\rho\end{aligned} \tag{80}$$

c. Total acoustic pressure

$$(p-p_o)(\bar{X},t,\theta_o) = (p-p_o)^{(b)}(\bar{X},t,\theta_o) + (p-p_o)^{(s)}(\bar{X},t,\theta_o) \tag{81}$$

where

$$\bar{X} = x_i e_{x_i} = X\cos\Theta\, e_{x_1} + X\sin\Theta\cos\Phi\, e_{x_2} + X\sin\Theta\sin\Phi\, e_{x_3}$$

$$\bar{Y} = (\lambda_i\phi - \lambda_p V_F \tau)\, e_{x_1} + \rho\cos(\phi+\theta_o+\delta_K-\tau)\, e_{x_2} + \rho\sin(\phi+\theta_o+\delta_K-\tau)\, e_{x_3}$$

$$\bar{R} = \bar{X} - \bar{Y} = r_i\, e_{x_i}$$

$$R = |\bar{R}| = X\left\{1 - \frac{2\rho}{X}\sin\Theta\cos(\phi+\theta_o+\delta_K-\Phi-\tau) - \frac{2(\lambda_i\phi-\lambda_p V_F\tau)}{X}\cos\Theta + \frac{\rho^2}{X^2} + \left(\frac{\lambda_i\phi-\lambda_p V_F\tau}{X}\right)^2\right\}^{1/2}$$

$$\bar{V} = \frac{1}{\lambda_p}\left(\frac{\partial\bar{Y}}{\partial\tau}\right)_{\rho,\phi} = -V_F\, e_{x_1} + \frac{\rho}{\lambda_p}\sin(\phi+\theta_o+\delta_K-\tau)\, e_{x_2} - \frac{\rho}{\lambda_p}\cos(\phi+\theta_o+\delta_K-\tau)\, e_{x_3}$$

$$\bar{M} = \frac{V_F\bar{V}}{a_o} = -M_F\, e_{x_1} + \rho M_p\sin(\phi+\theta_o+\delta_K-\tau)\, e_{x_2} - \rho M_p\cos(\phi+\theta_o+\delta_K-\tau)\, e_{x_3}$$

$$\bar{a} = a_i\, e_{x_i} = \left(\frac{\partial\bar{V}}{\partial\tau}\right)_{\rho,\phi} = -\frac{\rho}{\lambda_p}\cos(\phi+\theta_o+\delta_K-\tau)\, e_{x_2} - \frac{\rho}{\lambda_p}\sin(\phi+\theta_o+\delta_K-\tau)\, e_{x_3}$$

$$M_R = \frac{\bar{R}}{R}\cdot\bar{M} = \frac{X}{R}\left\{-M_F\left(\cos\Theta - \frac{\lambda_i\phi-\lambda_p V_F\tau}{X}\right) + \rho M_p\sin\Theta\sin(\phi+\theta_o+\delta_K-\Phi-\tau)\right\}$$

$$C^{+} = 1 - M_R$$

$$M_p = \frac{R_p \Omega}{a_0}$$

$$\frac{\partial \overline{M}}{\partial \tau} = -\rho M_p \cos(\phi + \theta_0 + \delta_K - \tau)\, e_{x_2} - \rho M_p \sin(\phi + \theta_0 + \delta_K - \tau)\, e_{x_3}$$

$$\frac{\partial^2 \overline{M}}{\partial \tau^2} = -\rho M_p \sin(\phi + \theta_0 + \delta_K - \tau)\, e_{x_2} + \rho M_p \cos(\phi + \theta_0 + \delta_K - \tau)\, e_{x_3}$$

$$\frac{\partial M_R}{\partial \tau} = \frac{\overline{R}}{R} \cdot \frac{\partial \overline{M}}{\partial \tau} = \rho M_p \frac{X}{R}\left\{ \frac{\rho}{X} - \sin\Theta \cos(\phi + \theta_0 + \delta_K - \Phi - \tau) \right\}$$

$$\frac{\partial^2 M_R}{\partial \tau^2} = \frac{\overline{R}}{R} \cdot \frac{\partial^2 \overline{M}}{\partial \tau^2} = -\rho M_p \frac{X}{R} \sin\Theta \sin(\phi + \theta_0 + \delta_K - \Phi - \tau)$$

$$\frac{\partial \overline{a}}{\partial \tau} = \frac{\partial^2 \overline{V}}{\partial \tau^2} = -\frac{\rho}{\lambda_p}\left\{ \sin(\phi + \theta_0 + \delta_K - \tau)\, e_{x_2} - \cos(\phi + \theta_0 + \delta_K - \tau)\, e_{x_3} \right\}$$

$$\overline{f} = V^*(\rho)\left\{ e_n - \epsilon(\rho)\, e_s \right\} = f_i\, e_{x_i} = \frac{V^*(\rho)}{\sqrt{\rho^2 + \lambda_i^2(\rho)}} \left\{ (\rho - \epsilon(\rho)\lambda_i(\rho))\, e_{x_1} \right.$$

$$\left. + (\lambda_i(\rho) + \rho\,\epsilon(\rho))\left(\sin(\phi + \theta_0 + \delta_K - \tau)\, e_{x_2} - \cos(\phi + \theta_0 + \delta_K - \tau)\, e_{x_3} \right) \right\}$$

$$\frac{\partial \overline{f}}{\partial \tau} = -\frac{V^*(\rho)}{\sqrt{\rho^2 + \lambda_i^2(\rho)}} (\lambda_i(\rho) + \rho\,\epsilon(\rho)) \left\{ \cos(\phi + \theta_0 + \delta_K - \tau)\, e_{x_2} \right.$$

$$\left. + \sin(\phi + \theta_0 + \delta_K - \tau)\, e_{x_3} \right\}$$

$$f_R = \frac{\overline{R}}{R} \cdot \overline{f} = \frac{X\, V^*(\rho)}{R\sqrt{\rho^2 + \lambda_i^2(\rho)}} \left\{ (\rho - \epsilon(\rho)) \left(\cos\Theta - \frac{\lambda_i \phi - \lambda_p V_F \tau}{X} \right) \right.$$

$$\left. + (\lambda_i(\rho) + \rho\,\epsilon(\rho)) \sin\Theta \sin(\phi + \theta_0 + \delta_K - \Phi - \tau) \right\}$$

$$\frac{\partial f_R}{\partial \tau} = \frac{\overline{R}}{R} \cdot \frac{\partial \overline{f}}{\partial \tau}$$

$$= \frac{X\, V^*(\rho)}{R\sqrt{\rho^2 + \lambda_i^2(\rho)}} (\lambda_i(\rho) + \rho\,\epsilon(\rho)) \left\{ \frac{\rho}{X} - \sin\Theta \cos(\phi + \theta_0 + \delta_K - \Phi - \tau) \right\}$$

It is noticed that the acoustic density perturbation has been replaced by the acoustic pressure, which is defined as the variation from atmospheric pressure. It is also noted that only steady sources are considered. Steady sources are those whose strength does not vary with time when viewed in the propeller-fixed coordinate system.

A few remarks concerning various aspects of Eqs. (79) and (80) are in order. The notation $[\ ]_{\tau=\tau_e}$ is used throughout this study to indicate that the quantity enclosed within the brackets is to be evaluated at source point (ρ, ϕ) and retarded time $\tau = \tau_e(\bar{X}, t, \rho, \phi)$ which is obtained by solving

$$G(\tau_e, t, \bar{X}, \rho, \phi) = \tau_e - t + M_p \left| \bar{X} - \bar{Y}(\tau_e, \rho, \phi) \right| \tag{82}$$

It is evident that Eq. (82) is an implicit equation for the required value of τ_e . If more than one solution to this equation exists (as it does at supersonic speed), each term in Eqs. (79) and (80) should be interpreted as a sum over all such solutions. At subsonic speed it has only one solution; for each source there is only one time at which it can transmit a signal to arrive at a given observer time, t. The next remark concerns the Doppler factor $C^{+} = 1 - \frac{\bar{R}}{R} \cdot \bar{M}$, which occurs in the denominator of each term in Eqs. (79) and (80). For supersonically moving sources, this factor can vanish at some point on the body and introduce a singularity with the resultant emission of Mach waves.

The difficulty in evaluation of the integrals for arbitrary moving sources is associated with the determination of retarded time for a given observer time, t. This difficulty may be avoided by Fourier analysis as

is done in most work concerning rotational noise. Since we are concerned with the total acoustic pressure, Fourier analysis will not be applied. Instead, numerical iteration is required.

Alternate forms of Eqs. (79) and (80)

For the aeroacoustic study of propeller noise, the following alternates to Eqs. (79) and (80) are presented:

Applying the chain rule to Eq. (82) shows that

$$\left\{ \left(\frac{\partial G}{\partial x_j}\right)_{\tau_e} + \left(\frac{\partial G}{\partial \tau_e}\right)_{\bar{X}} \frac{\partial \tau_e}{\partial x_j} \right\}_{\rho,\phi} = 0 \tag{83}$$

Introducing the Doppler factor C^+, we have

$$\left(\frac{\partial G}{\partial \tau_e}\right)_{\bar{X},\rho,\phi} = 1 - M_p \frac{\bar{R}}{R} \cdot \left(\frac{\partial \bar{Y}}{\partial \tau_e}\right)_{\bar{X},\rho,\phi} = C^+ \tag{84}$$

Upon using Eqs. (82) and (84) to eliminate G from Eq. (83), we find that

$$\left(\frac{\partial \tau_e}{\partial x_j}\right)_{\rho,\phi} = -\left[\frac{r_j M_p}{R\, C^+}\right]_{\tau=\tau_e} \tag{85}$$

Hence, applying the chain rule to an arbitrary function $f(\bar{X}, \tau_e)$ shows that

$$\frac{\partial}{\partial x_i}\left[f(\bar{X},\tau)\right]_{\tau=\tau_e} = \left[\frac{\partial f(\bar{X},\tau)}{\partial x_i} - \frac{r_i M_p}{R\, C^+} \frac{\partial f(\bar{X},\tau)}{\partial \tau}\right]_{\tau=\tau_e} \tag{86}$$

derivative with respect to x_i of a term of the form

$$\left[\frac{A(\tau)}{R|C^+|}\right]_{\tau=\tau_e}$$

or

$$\left[\operatorname{sign} C^+\left(\frac{A(\tau)}{R\,C^+}\right)\right]_{\tau=\tau_e}$$

where dependence on ρ and ϕ has been suppressed in order to simplify the notation.

Upon tedious mathematical manipulation, it is found that

$$\begin{aligned}\frac{\partial}{\partial x_i}\left[\frac{A(\tau)}{R|C^+|}\right]_{\tau=\tau_e} = &-\left[\operatorname{sign} C^+\left\{\frac{M_p r_i}{R^2 C^{+2}}\left(\frac{\partial A(\tau)}{\partial \tau}+\frac{A(\tau)}{C^+}\frac{\partial M_R}{\partial \tau}\right)\right\}\right]_{\tau=\tau_e} \\ &-\left[\operatorname{sign} C^+\left\{\frac{1}{R^2 C^{+2}}\left(-A(\tau)M_i+\frac{r_i A(\tau)(1-M^2)}{R\,C^+}\right)\right\}\right]_{\tau=\tau_e}\end{aligned} \tag{87}$$

$$\begin{aligned}\frac{\partial^2}{\partial x_i \partial x_j}\left[\frac{A(\tau)}{R|C^+|}\right]_{\tau=\tau_e} = &\left[\operatorname{sign} C^+ \frac{M_p^2 r_i r_j}{R^3 C^{+3}}\left\{\frac{\partial^2 A(\tau)}{\partial \tau^2}+\frac{A(\tau)}{C^+}\frac{\partial^2 M_R}{\partial \tau^2}\right.\right. \\ &\left.\left.+\frac{3}{C^+}\frac{\partial A(\tau)}{\partial \tau}\frac{\partial M_R}{\partial \tau}+\frac{3A(\tau)}{C^{+2}}\left(\frac{\partial M_R}{\partial \tau}\right)^2\right\}\right]_{\tau=\tau_e}+\left[-\operatorname{sign} C^+\left(\frac{M_p}{R^2 C^{+2}}\right)\left\{\frac{\partial A(\tau)}{\partial \tau}\left(\delta_{ij}\right.\right.\right. \\ &\left.+\frac{2}{C^+}\left(\frac{r_i M_j + r_j M_i}{R}\right)-\frac{3 r_i r_j (1-M^2)}{R^2 C^{+2}}\right)+\frac{A(\tau)}{C^+}\frac{\partial M_R}{\partial \tau}\left(\delta_{ij}+\frac{3}{C^+}\left(\frac{r_i M_j + r_j M_i}{R}\right)\right. \\ &\left.\left.\left.-\frac{6 r_i r_j (1-M^2)}{R^2 C^{+2}}\right)+\frac{A(\tau)}{C^+}\left(\frac{r_j}{R}\frac{\partial M_i}{\partial \tau}+\frac{r_i}{R}\frac{\partial M_j}{\partial \tau}+\frac{3 r_i r_j M_K}{R^2 C^+}\frac{\partial M_K}{\partial \tau}\right)\right\}\right]_{\tau=\tau_e} \\ &+\left[-\operatorname{sign} C^+\left(\frac{A(\tau)}{R^3 C^{+3}}\right)\left\{(1-M^2)\delta_{ij}+\frac{3(1-M^2)(r_i M_j + r_j M_i)}{R\,C^+}\right.\right. \\ &\left.\left.-\frac{3 r_i r_j (1-M^2)^2}{R^2 C^{+2}}-2M_i M_j\right\}\right]_{\tau=\tau_e}\end{aligned} \tag{88}$$

The terms inside the first bracket of each of Eqs. (87) and (88) represent the far field sound radiation from a point acoustic source while the rest of the terms represent the near field. It is interesting to note that the terms inside the third bracket of Eq. (88) dominate in the near field.

Thus, taking the $\overline{X}$ derivatives under the integral signs of Eqs. (79) and (80), we have the alternate forms

Force Noise

$$(p-p_o)^{(b)}(\overline{X},t,\theta_o)=\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} A^{(b)}(\overline{X},t,\rho,\phi,\theta_o)\,\gamma(\rho,\phi)\sqrt{\rho^2+\lambda_i^2(\rho)}\,d\phi\,d\rho \tag{89}$$

Thickness Noise

$$(p-p_o)^{(s)}(\overline{X},t,\theta_o)=\int_{\rho=r_h}^{1}\int_{\phi=\theta_L(\rho)}^{\theta_T(\rho)} A^{(s)}(\overline{X},t,\rho,\phi,\theta_o)\,t(\rho,\phi)\sqrt{\rho^2+\lambda_i^2(\rho)}\,d\phi\,d\rho \tag{90}$$

Total Noise

$$(p-p_o)(\overline{X},t,\theta_o)=(p-p_o)^{(b)}(\overline{X},t,\theta_o)+(p-p_o)^{(s)}(\overline{X},t,\theta_o) \tag{91}$$

where

$$A^{(b)}(\bar{X},t,\rho,\phi,\theta_0) = \frac{1}{4\pi}\sum_{K=1}^{B}\Big[\operatorname{sign} C^{+}\Big\{\frac{M_p}{RC^{+2}}\Big(\frac{\partial f_R}{\partial \tau} + \frac{f_R}{C^{+}}\frac{\partial M_R}{\partial \tau}\Big) - \frac{1}{R^2 C^{+2}}\Big(f_i M_i - \frac{f_R(1-M^2)}{C^{+}}\Big)\Big\}\Big]_{\tau=\tau_e} \tag{92}$$

$$A^{(s)}(\bar{X},t,\rho,\phi,\theta_0) = \frac{1}{4\pi}\sum_{K=1}^{B}\Big[\frac{\operatorname{sign} C^{+}}{R\lambda_p^2 C^{+4}}\Big\{\Big(\frac{\partial^2 M_R}{\partial \tau^2} + \frac{3}{C^{+}}\Big(\frac{\partial M_R}{\partial \tau}\Big)^2\Big) - \frac{1}{RM_p}\Big(3M_i\frac{\partial M_i}{\partial \tau} - \frac{1}{C^{+}}(1-6M^2+4M_R + M_R^2)\frac{\partial M_R}{\partial \tau}\Big) - \frac{1}{R^2C^{+}M_p^2}\Big((1+4M_R+M_R^2)M^2 - 3M^4 - 3M_R^2\Big)\Big\}\Big]_{\tau=\tau_e} \tag{93}$$

It is noticed that the dependence of acoustic pressure on θ_0, which is the angular displacement of the y-axis with respect to the x_2 axis at t = 0, has been expressed explicitly in Eqs. (89), (90), and (91). Thus, for a given location, $\bar{X}$, of the observer at time t, the instantaneous acoustic pressure depends on the initial angular displacement θ_0, of the propeller-fixed coordinate system.

E-3 Minimum Noise Criteria

Since the initial angular displacement θ_0 of the propeller-fixed coordinate system relative to the space-fixed coordinate system cannot be controlled by the experimenter, it is treated as a random variable. For equally spaced and identical blades, the probability density function of this random variable may be expressed as

$$f(\theta_0) = \frac{1}{2\pi} \qquad a \leq \theta_0 < a + 2\pi$$
$$= 0 \qquad \text{otherwise} \tag{94}$$

where a is an arbitrary constant. In this case, the acoustic pressure $(p - p_0)(\overline{X}, t, \theta_0)$ for a given $\overline{X}$ represents the entire family or ensemble of possible time histories which might have been the outcome of the same experiment. Since $(p - p_0)^2(\overline{X}, t, \theta_0)$ is a continuous function of the random variable θ_0, then $(p - p_0)^2(\overline{X}, t, \theta_0)$ is a Borel function of θ_0. From probability theory, the expectation of the random variable $(p - p_0)^2(\overline{X}, t, \theta_0)$ is equal to the expectation of the function $(p - p_0)^2(\overline{X}, t, \theta_0)$ with respect to the random variable θ_0

$$E\left((p-p_0)^2(\overline{X},t,\theta_0)\right) = \int_{-\infty}^{+\infty} (p-p_0)^2(\overline{X},t,\theta_0)\, f(\theta_0)\, d\theta_0 \tag{95}$$

which defines the mean square of $(p - p_0)(\overline{X}, t, \theta_0)$.

Substituting Eq. (94) into Eq. (95), we have

$$E\left((p-p_0)^2(\overline{X},t,\theta_0)\right) = \frac{1}{2\pi}\int_{a}^{2\pi+a} (p-p_0)^2(\overline{X},t,\theta_0)\, d\theta_0 \tag{96}$$

Letting a = Φ - t and $\theta_0 = \phi_0 + \Phi + t$ and substituting into Eq. (96), we have

$$E\left((p-p_0)^2(\bar{X},t,\theta_0)\right) = \frac{1}{2\pi}\int_0^{2\pi}(p-p_0)^2(\bar{X},t,\theta_0=\phi_0+\Phi+t)\,d\phi_0 \tag{97}$$

From Eq. (91) and all related definitions, it can easily be shown that

$$E\left((p-p_0)^2(\bar{X},t,\theta_0)\right) = \frac{B}{2\pi}\int_0^{\frac{2\pi}{B}}(p-p_0)^2(\bar{X},t,\theta_0=\phi_0+\Phi+t)\,d\phi_0 \tag{98}$$

It is easy to see from Eq. (98) that the mean square of the acoustic pressure for a stationary propeller is equal to its temporal mean square.

Before proceeding further, it may be stated that Eqs. (25), (26), (27), (48)(or 77), (59), (65)(or 68), (66)(or 69), and (98) are the basic equations of the aerodynamics and acoustic (aeroacoustic) propeller problem. Most propeller problems are generated from this set of equations.

A rather general problem of the design of a propeller for minimum noise is of finding the loading distribution and thickness distribution of a propeller blade that minimizes the ensemble mean square of the acoustic pressure subject to constraints on the aerodynamic performance. From the calculus of variations, such a problem may be mathematically represented by a nonlinear singular integro-differential Euler equation. So far, however, no one has succeeded in analyzing this problem. We shall consider the case for which the blade loading is modelled with a lifting

line and for which the thickness effects are ignored.

Following are the equations for this simplified propeller model:

Ideal thrust and power coefficients

$$C_{Ti} = 4B \int_{r=r_h}^{1} \Gamma(r) \left\{ \frac{r}{\lambda_p} + u_t^*(r) \right\} dr \tag{99}$$

$$C_{Pi} = \frac{4B}{\lambda_p} \int_{r=r_h}^{1} \Gamma(r) \left\{ V_F + u_a^*(r) \right\} r \, dr \tag{100}$$

Hydrodynamic advance coefficient

$$\lambda_i(r) = \frac{r\left\{ V_F + u_a^*(r) \right\}}{\frac{r}{\lambda_p} + u_t^*(r)} \tag{101}$$

Acoustic pressure (at t = 0, $\Phi = 0$)

$$(p - p_0)(X, \Theta, \phi_0) = \int_{\rho = r_h}^{1} K(X, \Theta, \rho, \phi_0) \, \Gamma(\rho) \, d\rho \tag{102}$$

where

$$K(X, \Theta, \rho, \phi_0) = \sum_{K=1}^{B} \left[\frac{\mathrm{sign}\, C^+}{2R\, C^{+2}} \left\{ M_p \left(\frac{\partial f_R}{\partial \tau} + \frac{f_R}{C^+} \frac{\partial M_R}{\partial \tau} \right) - \frac{1}{R} \left(f_i M_i - \frac{(1 - M^2) f_R}{C^+} \right) \right\} \right]_{\tau = \tau_e} \tag{103}$$

$$\tau_e = -R\dot{M}_p$$

$$R = X\left\{1 - \frac{2\rho}{X}\sin\Theta\cos(\phi_o + \delta_K - \tau_e) - \frac{2RM_F}{X}\cos\Theta + \frac{\rho^2}{X^2} + \frac{R^2M_F^2}{X^2}\right\}^{1/2} \tag{104}$$

$$f_R = \frac{X}{R}\left\{\left(\cos\Theta - \frac{RM_F}{X}\right)\left(\frac{\rho}{\lambda_p} + u_t^*\right) + \sin\Theta\sin(\phi_o + \delta_K - \tau_e)\,(V_F + u_a^*(\rho))\right\} \tag{105}$$

$$\frac{\partial f_R}{\partial \tau} = \frac{X}{R}(V_F + u_a^*(\rho))\left(\frac{\rho}{X} - \sin\Theta\cos(\phi_o + \delta_K - \tau_e)\right) \tag{106}$$

$$M_R = \frac{X}{R}\left\{-M_F\left(\cos\Theta + \frac{\lambda_p V_F \tau_e}{X}\right) + \rho M_p \sin\Theta\sin(\phi_o + \delta_K - \tau_e)\right\} \tag{107}$$

$$\frac{\partial M_R}{\partial \tau} = \rho M_p \frac{X}{R}\left\{\frac{\rho}{X} - \sin\Theta\cos(\phi_o + \delta_K - \tau_e)\right\} \tag{108}$$

$$M_i f_i = -M_F\left(\frac{\rho}{\lambda_p} + u_t^*(\rho)\right) + \rho M_p (V_F + u_a^*(\rho)) \tag{109}$$

$$M^2 = M_p^2 (V_F^2 \lambda_p^2 + \rho^2) \tag{110}$$

$$C^+ = 1 - M_R \tag{111}$$

Ensemble mean square of acoustic pressure (at t = 0, Φ = 0)

$$E\left((p-p_0)^2 (X, \Theta, \phi_0) \right) = \int_0^{\frac{2\pi}{B}} \frac{B}{2\pi} (p-p_0)^2 (X, \Theta, \phi_0) \, d\phi_0 \qquad (112)$$

It should be noticed that no loss in generality results from computing the ensemble mean square of acoustic pressure at t = 0 and Φ = 0 since the ensemble mean square of the acoustic pressure is independent of Φ and the dependence of the acoustic pressure on time, t, may be replaced by the dependence on X and Θ .

PART 2. NUMERICAL FORMULATION OF OPTIMUM NOISE PROPELLER PROBLEM FOR THE SIMPLIFIED MODEL

This part is concerned with the numerical formulation of the optimum noise propeller problem for the simplified model, and the computation of the lifting-line induced velocities. The study of aerodynamic problems of the propeller and of the "complex method" for constrained optimization had led to a nonlinear programming treatment of this model. We begin by expending the aerodynamic and acoustic parameters in terms of Chebyshev polynomials and bivariate Chebyshev polynomials (31, 1973).

The following new variables are introduced

$$\xi(r) = \frac{2r - r_h - 1}{1 - r_h} \quad ; \quad \xi_0(\rho) = \frac{2\rho - r_h - 1}{1 - r_h} \tag{113}$$

We note that

$$\xi(1) = \xi_0(1) = 1 \quad ; \quad \xi(r_h) = \xi_0(r_h) = -1$$

and

$$\rho - r = \frac{1}{2}(1 - r_h)(\xi_0 - \xi) \tag{114}$$

Since $\Gamma(r)$ is continuous between the hub and the tip and vanishes at both ends, it may be approximated by a truncated expansion in Chebyshev polynomials

$$\Gamma(q) = \sqrt{1-q^2}\ \sum_{m=1}^{M} G_m\, U_{m-1}(q) \tag{115}$$

where

$$U_{m-1}(q) = \frac{\sin(m \cos^{-1} q)}{\sqrt{1-q^2}}$$

With the aid of the formula

$$q\, U_{m-1}(q) - (1-q^2)\, U'_{m-1}(q) = m\, T_m(q)$$

we make the immediate identification

$$\frac{d\Gamma(r)}{dr}\, dr = \frac{d\Gamma(q)}{dq}\, dq = \sum_{m=1}^{M} \frac{-m\, G_m}{\sqrt{1-q^2}}\, T_m(q)\ dq \tag{116}$$

where

$$T_m(q) = \cos m \cos^{-1} q$$

Let both the axial induction factor and the tangential induction factor be approximated by finite double Chebyshev series of degree N in both q and q_o of the forms

$$I_a(q, q_o) = \sum_{i=0}^{N} \sum_{j=0}^{N}{}'' \ h_{ij}^{a}\, T_i(q)\, T_j(q_o) \tag{117}$$

$$I_t(q, q_o) = \sum_{i=0}^{N} \sum_{j=0}^{N}{}'' \ h_{ij}^{t}\, T_i(q)\, T_j(q_o) \tag{118}$$

The interpolation is made over the points $(q_r^{(n+1)}, q_{os}^{(n+1)})$, where $q_r^{(n+1)}$ and $q_{os}^{(n+1)}$ are roots of $T_{n+1}(q) = 0$ and $T_{n+1}(q_o) = 0$, respectively. That is

$$q_r^{(n+1)} = \cos\left\{\frac{(2r+1)\pi}{2(n+1)}\right\} \qquad r = 0(1)n$$

$$q_{os}^{(n+1)} = \cos\left\{\frac{(2s+1)\pi}{2(n+1)}\right\} \qquad s = 0(1)n$$

The double primes in (117) and (118) indicate that the first term is $h_{00}^{a}/4$ and $h_{00}^{t}/4$ and that h_{i0}^{a} and h_{i0}^{t}, h_{0j}^{a} and h_{0j}^{t} are to be taken as $h_{i0}^{a}/2$ and $h_{i0}^{t}/2$ and $h_{0j}^{t}/2$ for $i > 0$, $j > 0$, respectively.

The coefficients h_{ij}^{a} and h_{ij}^{t} are determined by a biorthogonality property (31, 1973).

$$\sum_{r=0}^{n}\sum_{s=0}^{n} T_{ij}(q_r^{(n+1)}, q_{os}^{(n+1)})\, T_{k\ell}(q_r^{(n+1)}, q_{os}^{(n+1)})$$

$$
\begin{aligned}
&= \frac{(n+1)^2}{4} && i = k \neq 0,\ j = \ell \neq 0 \\
&= \frac{(n+1)^2}{2} && i = k \neq 0,\ j = \ell = 0 \ \text{ or } \ i = k = 0,\ j = \ell \neq 0 \\
&= (n+1)^2 && i = j = k = \ell = 0 \\
&= 0 && i = k,\ j \neq \ell \\
& && i \neq k \quad j = \ell
\end{aligned}
$$

so that

$$h_{ij}^{a} = \frac{4}{(n+1)^2}\sum_{r=0}^{n}\sum_{s=0}^{n} I_a(q_r^{(n+1)}, q_{os}^{(n+1)})\, T_{ij}(q_r^{(n+1)}, q_{os}^{(n+1)}) \qquad (119)$$

$$\overset{t}{h}_{ij} = \frac{4}{(n+1)^2} \sum_{r=0}^{n} \sum_{s=0}^{n} I_t\left(q_r^{(n+1)}, q_{os}^{(n+1)}\right) T_{ij}\left(q_r^{(n+1)}, q_{os}^{(n+1)}\right) \quad (120)$$

A backwards recurrence formula for the evaluation of the sums of Eqs (119) and (120) is given in Appendix B.

A. The Evaluation of Induced Velocities

Substituting Eqs. (117), (118), (113) and (116) into Eqs. (50) and (51) respectively, we have

$$u_a^*(q) = \sum_{m=1}^{M} \frac{m G_m}{1 - r_h} \sum_{i=0}^{N} {\sum_{j=0}^{N}}'' \overset{a}{h}_{ij} T_i(q) \int_{-1}^{+1} \frac{T_m(q_0) T_j(q_0)}{\sqrt{1-q_0^2}\,(q_0 - q)} dq_0 \quad (121)$$

$$u_t^*(q) = \sum_{m=1}^{M} \frac{m G_m}{1 - r_h} \sum_{i=0}^{N} {\sum_{j=0}^{N}}'' \overset{t}{h}_{ij} T_i(q) \int_{-1}^{+1} \frac{T_m(q_0) T_j(q_0)}{\sqrt{1-q_0^2}\,(q_0 - q)} dq_0 \quad (122)$$

Furthermore, applying the solution for a Cauchy singular integral of the form

$$\int_{-1}^{+1} \frac{T_m(q_0)}{(q_0 - q)\sqrt{1-q_0^2}} dq_0 = \pi U_{m-1}(q)$$

we obtain

$$u_a^*(q) = \sum_{m=1}^{M} \frac{\pi m G_m}{1 - r_h} \left\{ \sum_{i=0}^{N} {\sum_{j=0}^{m}}'' \overset{a}{h}_{ij} T_i(q) T_j(q) U_{m-1}(q) \right.$$

$$\left. + \sum_{i=0}^{N} {\sum_{j=m+1}^{N}}'' \overset{a}{h}_{ij} T_i(q) T_m(q) U_{j-1}(q) \right\}$$

$$= {\sum_{i=0}^{2N+M-1}}' u_{a,i} T_i(q) \quad (123)$$

$$u_t^*(q) = \sum_{m=1}^{M} \frac{\pi m G_m}{1 - r_h} \left\{ \sum_{i=0}^{N} \sum_{j=0}^{m}{}'' h_{ij}^{t} \, T_i(q) \, T_j(q) \, U_{m-1}(q) \right.$$

$$\left. + \sum_{i=0}^{N} \sum_{j=m+1}^{N}{}'' h_{ij}^{t} \, T_i(q) \, T_m(q) \, U_{j-1}(q) \right\}$$

$$= \sum_{i=0}^{2N+M-1}{}' u_{t,i} T_i(q) \tag{124}$$

where $u_{a,i}$ and $u_{t,i}$ are the coefficients of the Chebyshev expansions of the axial and tangential components of the induced velocity, respectively.

B. The Evaluation of Hydrodynamic Advance-Coefficients

Combining Eqs (123), (124) and (113), we have

$$\lambda_i(q) = \frac{r(q) \left(V_F + \sum_{i=0}^{2N+M-1}{}' u_{a,i} T_i(q) \right)}{\frac{r(q)}{\lambda_p} + \sum_{i=0}^{2N+M-1}{}' u_{t,i} T_i(q)} \tag{125}$$

It should be noticed that Eq. (125) is an implicit equation for the required value of $\lambda_i(q)$ for a given $\Gamma(q)$ since the induction factors are functions of $\lambda_i(q)$. Equation (125) will be used to obtain the new approximation for $\lambda_i(q)$ from the current value $\lambda_i(q)$ in the iterative scheme for solving the non-optimum propeller problem (8, 1952) to obtain an initial solution for the input of the complex method.

C. The Chebyshev Coefficients of Circulation

Let $q_l^{(M)}$ be the roots of $T_M(q) = 0$ and $\lambda_l^{(M)}$ the values of $\lambda_i(q)$ at $q_l^{(M)}$.

Using the Eq. (101), and Equations (123) and (124), we obtain a system of linear equations for M unknowns, G_m.

$$\frac{\lambda_\ell^{(M)}}{\lambda_p} - V_F = \sum_{m=1}^{M} \left\{ \sum_{i=0}^{N} {\sum_{j=0}^{m}}'' \left(h_{ij}^{a} - \frac{2\lambda_\ell^{(M)} h_{ij}^{t}}{(1-r_h)\vartheta_\ell^{(M)} + 1 + r_h} \right) \Lambda_{\ell ijmm}^{(M)} \right.$$

$$\left. + \sum_{i=0}^{N} {\sum_{j=m+1}^{N}}'' \left(h_{ij}^{a} - \frac{2\lambda_\ell^{(M)} h_{ij}^{t}}{(1-r_h)\vartheta_\ell^{(M)} + 1 + r_h} \right) \Lambda_{\ell imjm}^{(M)} \right\} G_m$$

$$\ell = 1(1)M \qquad (126)$$

where h_{ij}^{a} and h_{ij}^{t} are obtained from Eqs. (119) and (120), respectively, and

$$\Lambda_{\ell ijkm}^{(M)} = \frac{\pi m}{1-r_h} T_i(\vartheta_\ell^{(M)}) T_j(\vartheta_\ell^{(M)}) U_{k-1}(\vartheta_\ell^{(M)}) \qquad (127)$$

The coefficients G_m are determined by solving Eqs. (126).

D. The Evaluation of Thrust and Power Coefficients

Using Eqs (99) and (100) and the orthogonality property of the Chebyshev Polynomials, we obtain expressions for the ideal thrust and power coefficients:

$$C_{Ti} = \frac{B(1-r_h)\pi}{2} \left\{ \frac{1+r_h}{\lambda_p} G_1 + \frac{1-r_h}{2\lambda_p} G_2 \right.$$

$$\left. + \sum_{m=1}^{M} G_m \left(u_{t,m-1} - u_{t,m+1} \right) \right\} \qquad (128)$$

$$C_{pi} = \frac{B(1-r_h)\pi}{2\lambda_p} \Big\{ G_1(1+r_h)V_F + G_2 \frac{(1-r_h)}{2} V_F$$

$$+ \sum_{m=1}^{M} \frac{(1+r_h)}{2} G_m \left(u_{a,m-1} - u_{a,m+1} \right)$$

$$+ \sum_{m=1}^{M} \frac{1-r_h}{4} G_m \left(u_{a,|m-2|} - u_{a,m+2} \right) \qquad (129)$$

E. The Evaluation of Acoustic Pressure

To compute the instantaneous acoustic pressure, we separate the kernel $K(X, \Theta, \rho, \phi_0)$ of Eq. (102) into three parts

$$K(X,\Theta,\rho,\phi_0) = K_0(X,\Theta,\rho,\phi_0)$$
$$+ K_a(X,\Theta,\rho,\phi_0)\, u_a^*(\rho) + K_t(X,\Theta,\rho,\phi_0)\, u_t^*(\rho) \qquad (130)$$

where

$$K_0(X,\Theta,\rho,\phi_0) = \sum_{K=1}^{B} \Bigg[\frac{1}{2RC^{+2}} \Big\{ M_p V_F \Big(\frac{\rho}{R} -$$

$$\frac{X}{R} \sin\Theta \cos(\phi_0 + \delta_K + M_p R) \Big) + \frac{1}{C^+} \Big(\frac{1-M^2}{R} + M_p \frac{\partial M_R}{\partial \tau} \Big) \Big($$

$$\frac{\rho}{\lambda_p} \Big(\frac{X}{R} \cos\Theta - M_F \Big) + V_F \frac{X}{R} \sin\Theta \sin(\phi_0 + \delta_K + M_p R) \Big) \Big\} \Bigg]_{\tau=\tau_e} \qquad (131)$$

$$K_a(X,\Theta,\varrho,\phi_o) = \sum_{k=1}^{B}\left[\frac{1}{2RC^{+2}}\left\{-\frac{X}{R}M_p \sin\Theta \cos(\phi_o+\delta_k+M_pR)\right.\right.$$

$$\left.\left. + \frac{1}{C^+}\left(\frac{1-M^2}{R} + M_p\frac{\partial M_R}{\partial \tau}\right)\frac{X}{R}\sin\Theta \sin(\phi_o+\delta_k+M_pR)\right\}\right]_{\tau=\tau_e} \tag{132}$$

$$K_t(X,\Theta,\varrho,\phi_o) = \sum_{K=1}^{B}\left[\frac{1}{2RC^{+2}}\left\{\frac{M_F}{R} + \frac{1}{C^+}\left(\frac{1-M^2}{R} + \right.\right.\right.$$

$$\left.\left.\left. M_p\frac{\partial M_R}{\partial \tau}\right)\left(\frac{X}{R}\cos\Theta - M_F\right)\right\}\right]_{\tau=\tau_e} \tag{133}$$

Furthermore, let $K_o(X,\Theta,q_o,\phi_o)$, $K_a(X,\Theta,q_o,\phi_o)$, and $K_t(X,\Theta,q_o,\phi_o)$ be approximated by finite Chebyshev series of degree L in q_o of the form

$$K_o(X,\Theta,\varrho_o,\phi_o) = {\sum_{k=0}^{L}}' P_{o,k}(X,\Theta,\phi_o)\,T_k(\varrho_o) \tag{134}$$

$$K_a(X,\Theta,\varrho_o,\phi_o) = {\sum_{k=0}^{L}}' P_{a,k}(X,\Theta,\phi_o)\,T_k(\varrho_o) \tag{135}$$

$$K_t(X,\Theta,\varrho_o,\phi_o) = {\sum_{K=0}^{L}}' P_{t,K}(X,\Theta,\phi_o)\,T_K(\varrho_o) \tag{136}$$

where $P_{o,k}(X,\Theta,\phi_o)$, $P_{a,k}(X,\Theta,\phi_o)$, and $P_{t,k}(X,\Theta,\phi_o)$ can easily be obtained by using Eq. (B-2) of Appendix B.

The advantage of separation of the kernel is that the coefficients $P_{o,k}(X,\Theta,\phi_o)$, $P_{a,k}(X,\Theta,\phi_o)$, and $P_{t,k}(X,\Theta,\phi_o)$ may be computed once for all for given B, M_p, M_F, X,Θ, and ϕ_o since they are independent of the helixal vortex system behind the propeller.

Substituting Eqs. (130) and (115) together with Eqs. (134), (135),

and (136) into Eq. (102) we have

$$(p - p_0)(\chi, \Theta, \phi_0) = \frac{1 - r_k}{2} \left\{ \frac{\pi}{4} \sum_{m=1}^{M} G_m \left(P_{o,m-1} - P_{o,m+1} \right) + \sum_{m=1}^{M} G_m \sum_{K=0}^{L}{}' \left(P_{a,K} P_{a,m,K} + P_{t,K} P_{t,m,K} \right) \right\} \tag{137}$$

where

$$P_{a,m,K} = \int_{-1}^{+1} \sqrt{1 - \zeta_0^2}\, T_K(\zeta_0)\, U_{m-1}(\zeta_0)\, u_a^*(\zeta_0)\, d\zeta_0$$

$$= \frac{\pi}{8} \sum_{j=0}^{2N+M-1}{}' u_{a,j} \left(\delta_{j,0}\delta_{K,0}\delta_{m,1} + \delta_{j,K+m-1} + \delta_{j,0}\delta_{K,m-1} + \delta_{j,|K-m+1|} - \delta_{j,m+K+1} - \delta_{j,0}\delta_{K,m+1} - \delta_{j,|K-m-1|} \right) \tag{138}$$

$$P_{t,m,K} = \int_{-1}^{+1} \sqrt{1 - \zeta_0^2}\, T_K(\zeta_0)\, U_{m-1}(\zeta_0)\, u_t^*(\zeta_0)\, d\zeta_0$$

$$= \frac{\pi}{8} \sum_{j=0}^{2N+M-1}{}' u_{t,j} \left(\delta_{j,0}\delta_{K,0}\delta_{m,1} + \delta_{j,K+m-1} + \delta_{j,0}\delta_{K,m-1} + \delta_{j,|K-m+1|} - \delta_{j,m+K+1} - \delta_{j,0}\delta_{K,m+1} - \delta_{j,|K-m-1|} \right) \tag{139}$$

$$\delta_{i,j} = 0 \quad \text{if } i \neq j$$
$$= 1 \quad \text{if } i = j$$

F. The Evaluation of the Ensemble Mean Square of Acoustic Pressure

We start by writing Eq. (112) as

$$E\left((p-p_o)^2(X,\Theta,\phi_o)\right) = \frac{B}{2\pi}\int_0^{\frac{2\pi}{B}} (p-p_o)^2(X,\Theta,\phi_o)\,d\phi_o$$

$$= \frac{1}{2}\int_{-1}^{+1}(p-p_o)^2(X,\Theta,\eta)\,d\eta \qquad (140)$$

where

$$\phi_o = \frac{\pi}{B}(1+\eta)$$

Now, let η_j and W_j be the abscissas and weights of the K-point Gauss-Legendre formula. Then Eq. (140) may be approximated by

$$E\left((p-p_o)^2(X,\Theta,\phi_o)\right) = \frac{1}{2}\sum_{j=1}^{K} W_j\,(p-p_o)^2(X,\Theta,\eta_j) \qquad (141)$$

where $(p - p_o)(X,\Theta, \eta_j)$ are evaluated using Eq. (137).

It should be noticed that the retarded time and the distance R are computed by using Newton's method (see Appendix C).

G. The Nonlinear Programming for the Simplified Propeller Model

In this section we are concerned with the formulation of a nonlinear programming model for the simplified propeller. To begin, we assume that the number of propeller blades, B, the advance (or forward) Mach number, M_F, the tip Mach number, M_p, the distance between the observer and the center of the propeller, X, and the azimuth angle, Θ, of the observer are known.

Investigating the evaluation of the aerodynamic and acoustic quantities of the propeller, we find that for a given $\lambda_i(q_o)$, all these quantities are

determined. That is, for a given configuration of the helical vortex system behind the propeller all aerodynamic and acoustic characteristics are fixed. This suggests that it is possible to find a configuration which satisfies all the specified constraints and produces minimum noise. A nonlinear programming model is established to facilitate numerical determination of this optimum configuration.

Let $\lambda_j^{(J+1)}$ be the values of $\lambda_i(q_o)$ at the zeros of $T_{J+1}(q_o)$. A nonlinear programming model for the simplified propeller has the following form:

$$\text{Maximize} \quad E^*(\lambda_1^{(J+1)}, \lambda_2^{(J+1)}, \ldots, \lambda_{J+1}^{(J+1)}) = -E\left((p-\bar{p})^2(X, \Theta, \phi_o)\right)$$

$$\text{Subject to} \quad \begin{aligned} &\lambda_{L,j} \leq \lambda_j^{(J+1)} \leq \lambda_{U,j} \qquad j = 1(1)J+1 \\ &C_{TL} \leq C_{Ti} \leq C_{TU} \\ &C_{PL} \leq C_{Pi} \leq C_{PU} \end{aligned} \tag{142}$$

where the ideal thrust and power coefficients are regarded as implicit variables while $\lambda_1^{(J+1)}$, $\lambda_2^{(J+1)}$, $\lambda_{J+1}^{(J+1)}$ are the explicit independent variables. The upper and lower constraints are either constants or functions of the independent variables. It is noticed that the value of $\lambda_i(q_o)$ at any point q_o is interpolated from the polynomial interpolation of degree J which exactly fits $\lambda_i(q_o)$ at $q_j^{(J+1)}$, j = 1, 2, ... , J+1.

The algorithm of J. A. Richardson and J. L. Kuester (32, 1973) based on the "complex" method of M. J. Box (30, 1965) has been modified with two feasible starting points as input to solve the nonlinear programming (142). The constrained complex method is a sequential search technique. Since the initial

set of points is randomly scattered throughout the feasible region, the procedure should tend to find the global maximum. The first feasible starting point is the solution of the aerodynamic optimum propeller. The second feasible starting point is any solution for a non-optimum propeller. These two feasible solutions are generated by using the iterative technique given in Ref. 8. The procedures are described below.

A propeller having a constant hydrodynamic coefficient is called an aerodynamic optimum propeller since its ideal efficiency is, according to Betz, the greatest that can be obtained for a given propeller advance coefficient and ideal thrust coefficient.

Procedure for Finding Aerodynamic Optimum Solution

1. Specify the loading coefficient C_{Ti} (or C_{Pi}) which satisfies the implicit constraint of Eq. (142)
2. Assume $\lambda_{\ell}^{(M)} = \lambda_i$, $\ell = \ell(1)M$
3. Solve the system of Eqs. (126) for G_m
4. Compute C_{Ti} (or C_{Pi}) from Eq. (128) (or Eq. 129).

Repeating steps (2) through (4) for several values of λ_i, the dependence of the loading coefficients on λ_i is derived from which the proper value λ^* is interpolated. Thus, the first feasible solution is obtained by setting

$$\lambda_j^{(J+1)} = \lambda^* \qquad j = 1(1)\,J+1 \tag{143}$$

Procedure for Finding a Non-Optimum Solution

1. Specify the loading coefficient C_{Ti} (or C_{Pi}) which satifies the implicit constraint of Eq. (142)

2. Specify a characterising function for the circulation

$$g = \sqrt{1-q^2}\sum_{m=1}^{M} g_m U_{m-1}(q) \tag{144}$$

3. Relate the circulation $\Gamma(q)$ which satisfies the loading coefficient to the given characterising function by a factor k which is independent of q

$$\Gamma(q) = k g = k\sqrt{1-q^2}\sum_{m=1}^{M} g_m U_{m-1}(q) \tag{145}$$

or

$$G_m = k g_m$$

4. Assume $\lambda_j^{(J+1)} = \lambda_F \quad j = 1(1)\ J+1$

5. Compute $u_{a,i}$ and $u_{t,i}$ from Eqs. (123) and (124), respectively, by replacing G_m by g_m

$$u_a^*(q) = k\sum_{i=0}^{2N+M-1}{}' u_{a,i} T_i(q) \tag{146}$$

$$u_t^*(q) = k\sum_{i=0}^{2N+M-1}{}' u_{t,i} T_i(q) \tag{147}$$

6. Substitute Eqs. (145), (146), and (147) into Eq. (128) (or 129)

$$C_{Ti} = k C_{T1} + k^2 C_{T2} \tag{148}$$

or

$$C_{pi} = k C_{p1} + k^2 C_{p2} \tag{149}$$

where

$$C_{T1} = \frac{B(1-r_h)\pi}{2\lambda_p}\left(g_1(1+r_h) + g_2\frac{1-r_h}{2}\right)$$

$$C_{T2} = \frac{B(1-r_h)\pi}{2}\sum_{m=1}^{M} g_m(u_{t,m-1} - u_{t,m+1})$$

$$C_{P1} = \frac{BV_F(1-r_h)\pi}{2\lambda_p}\left(g_1(1+r_h) + g_2\frac{1-r_h}{2}\right)$$

$$C_{P2} = \frac{B(1-r_h)\pi}{4\lambda_p}\left\{(1+r_h)\sum_{m=1}^{M} g_m(u_{a,m-1} - u_{a,m+1}) + \frac{1-r_h}{2}\sum_{m=1}^{M} g_m(u_{a,|m-2|} - u_{a,m+1})\right.$$

7. Solve Eq. (148) (or 149) for k.

8. Obtain a new approximation set of $\lambda_j^{(J+1)}$ by substituting Eqs. (146) and (147) into Eq. (101)

Repeat steps (5) through (8) until a satisfactory feasible solution is obtained.

The numerical computation has been programmed for the CDC Cyber 175 computer. The FORTRAN listing for the program may be found in Appendix D. Details for use of the program is outlined in the main program. In addition to the nonlinear program Eq. (142), this program is capable of solving the following three problems:

Problem I:

Given: B, r_h, V_F, λ_p (or M_F and M_p), $\lambda_i(r) = \lambda_i$, X, and

Determine: Aerodynamic optimum circulation distribution,

Induced velocity components

C_{Ti}, C_{Pi}

Ideal efficiency

Ensemble mean square of the acoustic pressure (optional).

Problem II:

Given: B, r_h, V_F, λ_p (or M_F and M_p), $\lambda_i(r) = \lambda_i$,

C_{Ti} (or C_{Pi}), X, and $\textcircled{H}$

Determine: Aerodynamic optimum circulation distribution,

Induced velocity components,

Ideal efficiency

Ensemble mean square of the acoustic

pressure (optional)

Problem III:

Given: B, r_h, V_F, λ_p (or M_F and M_p),

C_{Ti} (or C_{Pi}), X, and $\textcircled{H}$

The type of characterising function of circulation

Determine: Circulation distribution,

Induced velocity components,

Ideal efficiency

Ensemble mean square of the acoustic

pressure (optional)

Results of sample calculations are present in Part 3.

Having determined the acoustic optimum circulation and the hydrodynamic advance coefficient distributions for the propeller, the actual shape of the blades remains to be determined by lifting-surface technique. It is only

necessary to select an appropriate chordwise circulation distribution and a thickness form at each blade section to evaluate the lifting-surface velocities. The mean line at each blade section is then obtained from Eq. (48). The reader is referred to Refs. 10, and 35 for detailed numerical computation.

PART 3. APPLICATIONS AND NUMERICAL RESULTS

The main purpose of the computing program based on the technique developed in Part 2 is to solve the nonlinear programming Eq. (142) to find the acoustic optimum circulation within prescribed aerodynamic constraints. In order to give some verification of the present technique, some computed examples are given in the following

Numerical Examples

1. Given: $B = 5$, $r_h = 0.2$, $\lambda_p = 0.19966$, $V_F = 1$, and $\lambda_i = 0.27211$.

   Determine: Aerodynamic optimum circulation distribution,
   induced velocity
   thrust and power coefficients,
   ideal efficiency

The results for M = 10 and N = 10 are shown in Table 1.

The last three columns in Table 1 show a comparison of results for the same propeller obtained from Ref. 10. The agreement between these methods supports the validity of the present aerodynamic model and computing program. Finally, the thrust and power coefficients have been determined, $C_{Ti} = 1.23324$, $C_{Pi} = 1.68071$. From these, the ideal efficiency is 0.73376. From the well-known relation for an optimum propeller, $\eta_i = \lambda_p V_F/\lambda_i$, there is obtained $\eta_i = 0.73375$.

Table 1

Results for an Optimum Free-Running

Five-Bladed Propeller with $\lambda_p = 0.19966$, $V_F = 1$, $\lambda_i = 0.27211$

$G_1 = 0.03301036$	$G_2 = 0.00341272$	$G_3 = 0.00125070$
$G_4 = 0.00018100$	$G_5 = -0.00005210$	$G_6 = -0.00001478$
$G_7 = -0.00000722$	$G_8 = 0.00000089$	$G_9 = 0.00000028$
$G_{10} = 0.00000027$		

$C_{Ti} = 1.23324$

$C_{Pi} = 1.68071$

$\eta_i = 0.73376$

r	q	u_a^*	u_t^*	Γ	Kerwin's vortex-line method Γ	Lerb's induction factor method Γ
0.2	-1.00	0.12674	-0.17197	0.0	0.0	0.0
0.3	-0.75	0.19899	-0.18047	0.01945	0.0197	0.0196
0.4	-0.05	0.24802	-0.16869	0.02582	0.0260	0.0258
0.5	-0.25	0.27992	-0.15233	0.02955	0.0297	0.0296
0.6	0.00	0.30098	-0.13650	0.03171	0.0318	0.0317
0.7	0.25	0.31522	-0.12253	0.03252	0.0325	0.0325
0.8	0.50	0.32521	-0.11062	0.03142	0.0314	0.0314
0.9	0.75	0.33240	-0.10050	0.02634	0.0262	0.0263
1.0	1.00	0.33719	-0.09154	0.0	0.0	0.0

2. Given: $B = 2$, $r_h = 0.2$, $V_F = 1$, $\lambda_p = 0.26932$, $M_F = 0.2$,

$M_p = 0.7426$, $X = 4.272$, and $\Theta = 1.212$ rad.

Constraints

$$0.28 \leq \lambda_i \leq 0.5$$

$$0.7467 \leq C_{Ti}$$

Determine: Acoustic optimum circulation distribution,

induced velocity

thrust and power coefficients

ideal efficiency

For this particular case only the feasible aerodynamic optimum solution is necessary to be used as the initial solution in the complex method. All points, which were generated from random numbers and constraints, converged to $\lambda_i = 0.28$ quite rapidly. The results are shown in Figs. (9) through (11). It is not surprising that the acoustic optimum solution is the aerodynamic optimum solution that has minimum thrust coefficient since all loadings in this case are similar as shown in Fig. 10. This provides a simple numerical check of the technique developed in Part 2.

It is noted that Fig. 11 is plotted based on the following definition of the total sound pressure level:

$$\text{Total Sound Pressure Level} = 10 \log_{10} \frac{E((p-p_0)^2)}{p_{ref}^2}$$

$$= 10 \log_{10} E((p-p_o)^2) + 20 \log_{10} \frac{\rho_o V_p^2}{p_{ref}} \text{ dB}$$

In order to apply the present technique to a more general case, the third example is chosen as follows

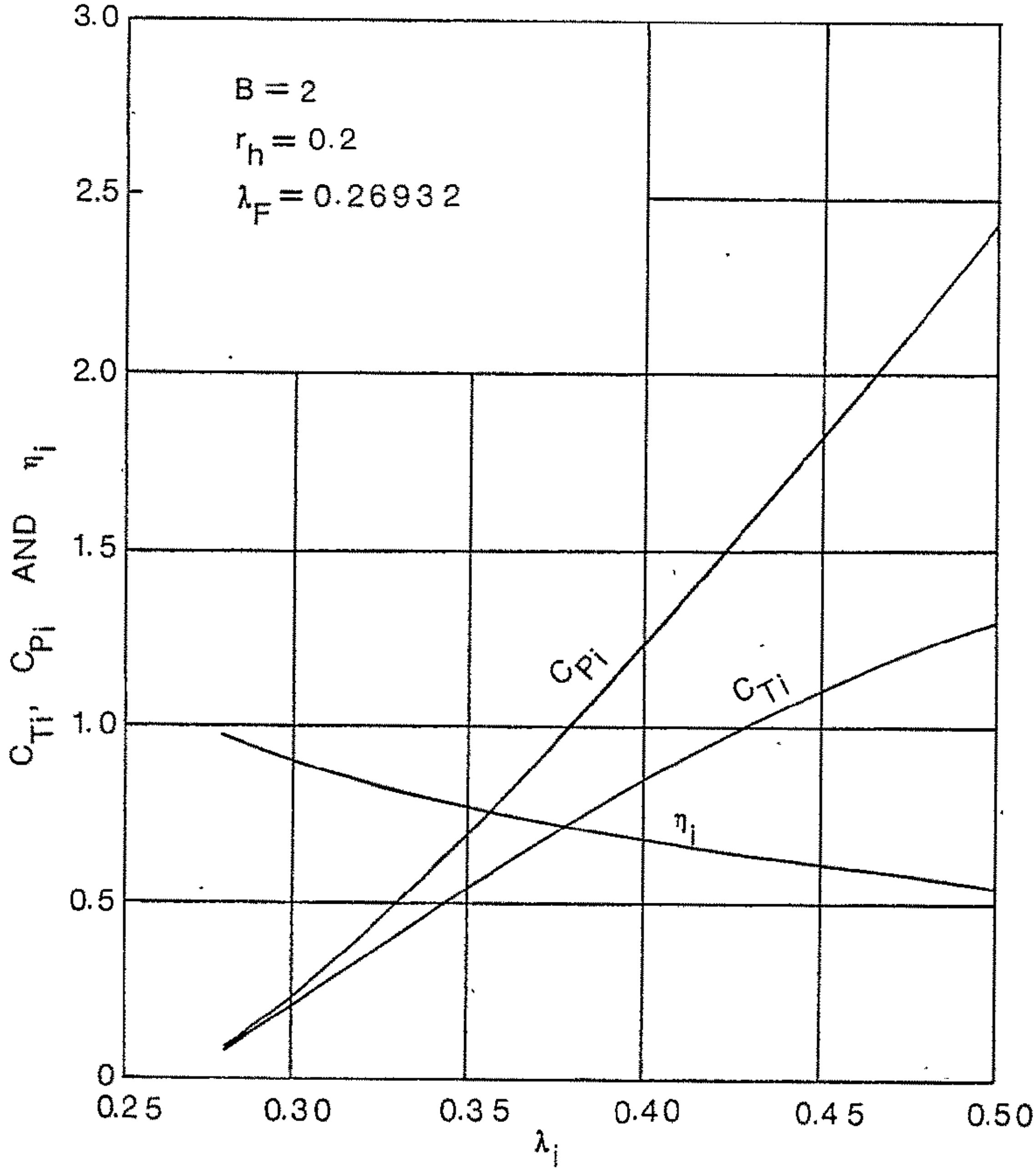

Fig. 9 Relation between C_{Ti}, C_{Pi}, η_i and λ_i.

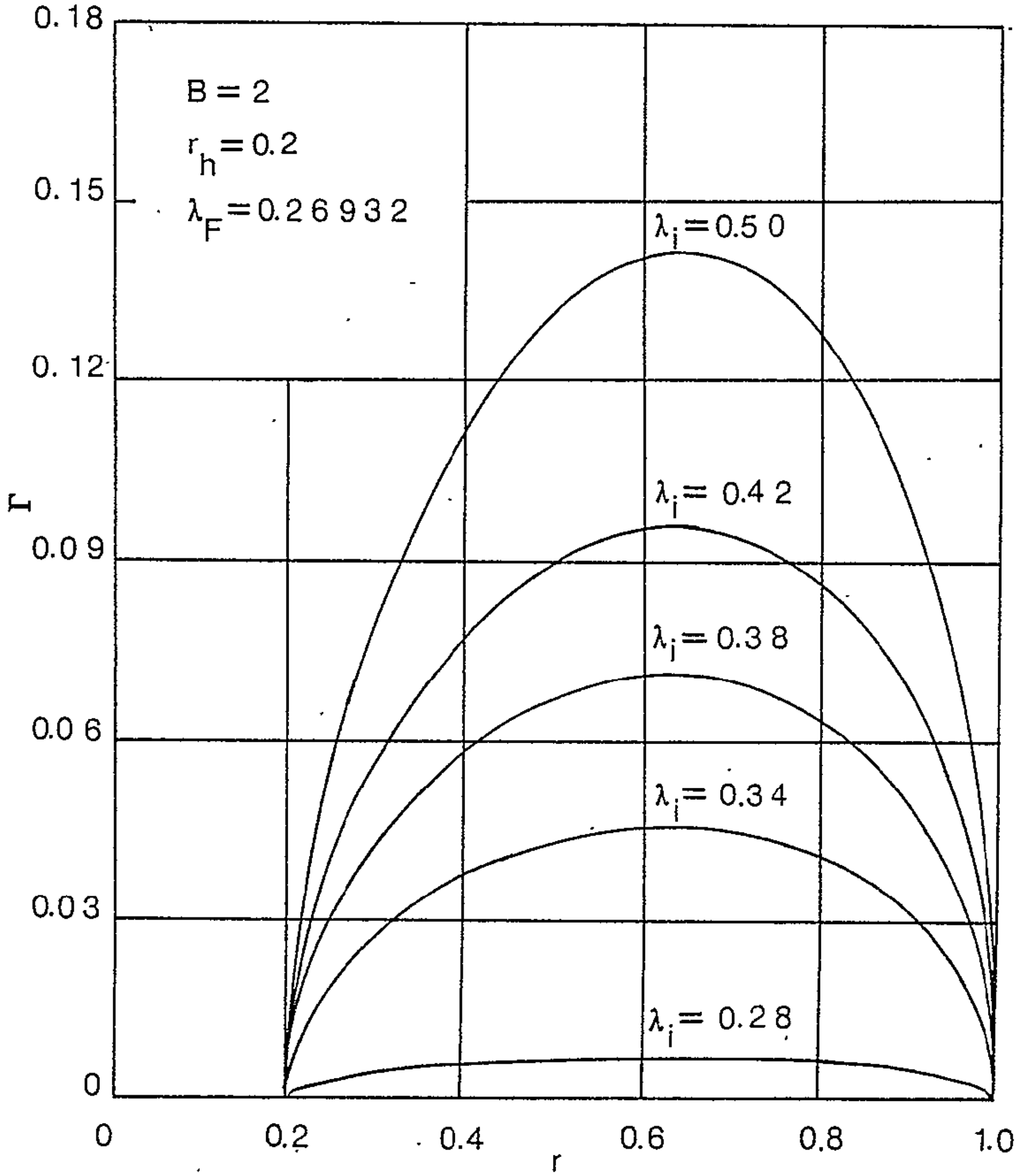

Fig. 10 Relation between aerodynamic optimum circulation and λ_i.

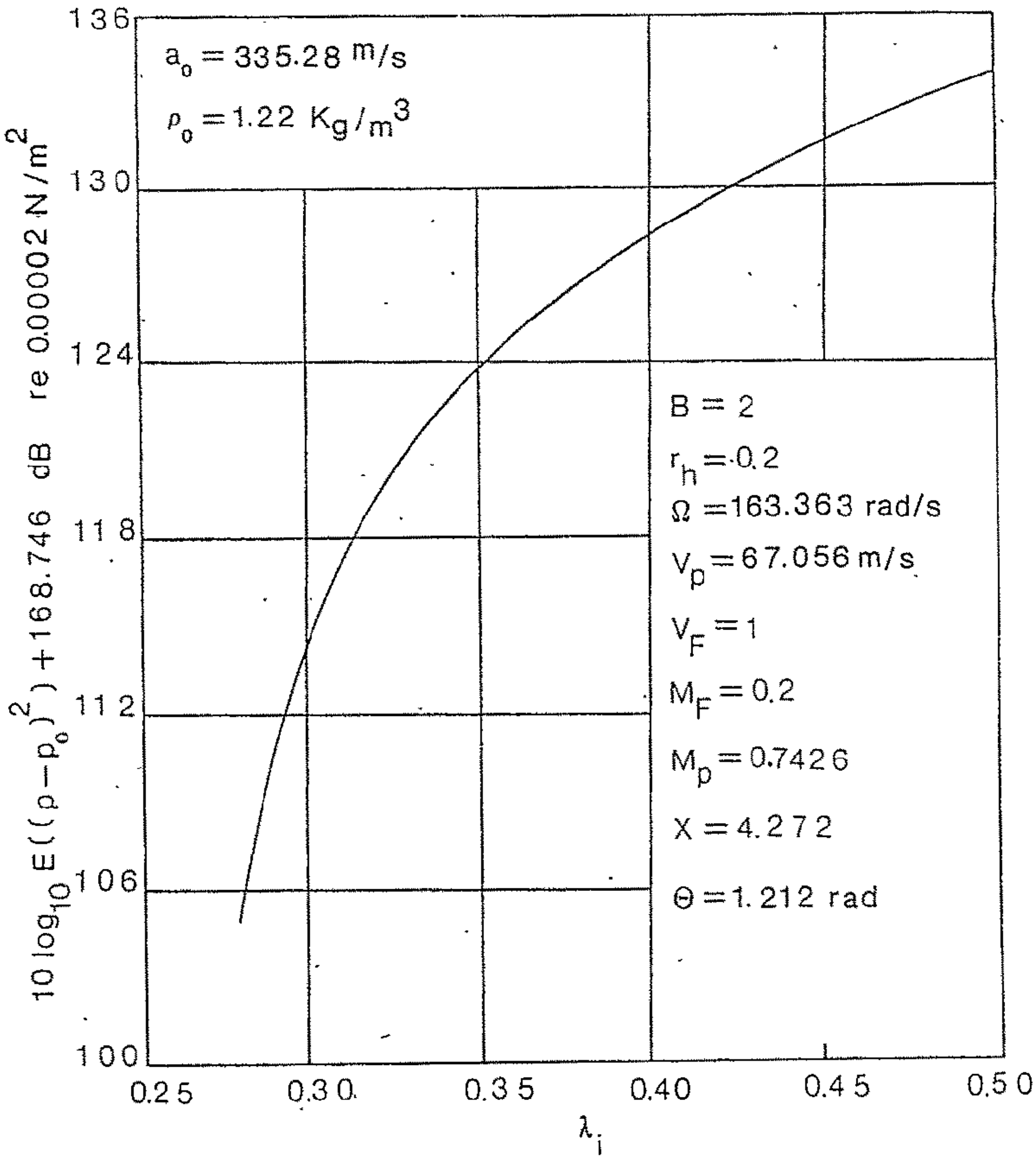

Fig. 11 Ensemble mean square of acoustic pressure for aerodynamic optimum propellers.

3. Given: $B = 2$, $R_p = 1.524$ m $= 0.2$, $V_p = 67.056$ m/s

$\Omega = 163.363$ rad/s, $V_F = 1$,

Total Thrust = 7116.8 N

Constraints

$$0.1 \leq \lambda_j^{(J+1)} \leq 0.6 \qquad j = 1(1)10$$

$$0.35560 \leq C_{Ti}$$

Determine: Acoustic optimum circulation distribution,

induced velocity,

thrust and power coefficients

ideal efficiency.

In calculation of the ensemble mean square of the acoustic pressure, the following values were used

air density $\rho_o = 1.22$ kg/m^3

speed of sound $a_o = 335.28$ m/s

In the numerical computation, J, L, M, and N were taken to be 9, 10, 10, and 10 respectively. The two feasible solutions corresponding to $C_{Ti} = 0.3556$ are shown in Fig. 12. The characterising function is also shown in Fig. 12

$$g(q) = 0.03267 - 0.00083\ U_1(q) - 0.00467\ U_2(q)$$
$$- 0.00243\ U_3(q) - 0.00027\ U_4(7)$$

After 19 iterations the factor k was found to be 1.01916 and all $\lambda_j^{(J+1)}$ satisfied the convergence criterion

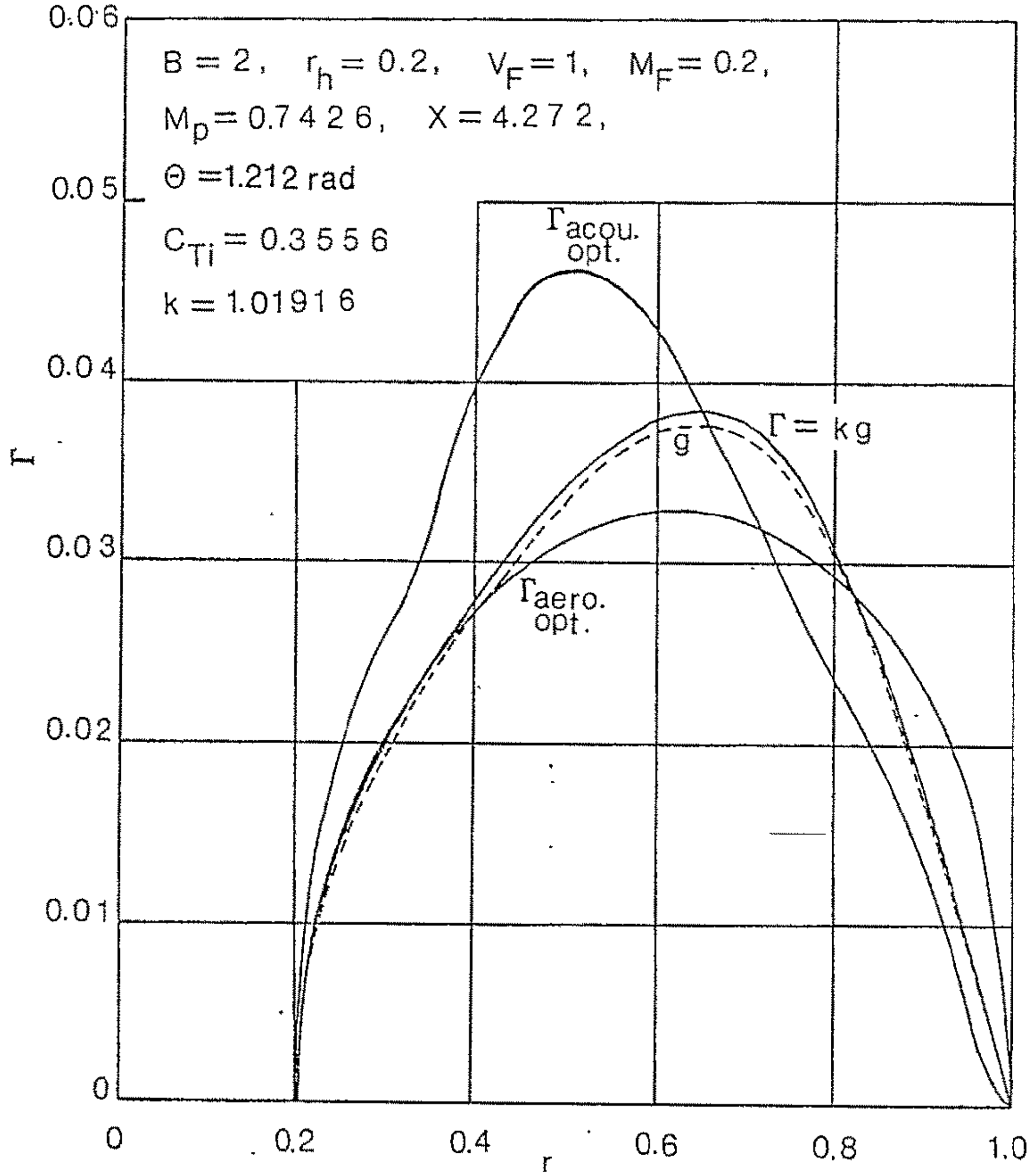

Fig. 12 Acoustic optimum and feasible circulations.

$$\left| \lambda_{j,\, n+1}^{(J+1)} - \lambda_{j,n}^{(J+1)} \right| \leq 10^{-6}$$

The acoustic optimum solution was found after 117 iterations. Convergence for the complex method was assumed when the objective function values at each point were within 10^{-8} units for 4 consecutive iterations. The toral execution time was 164.531 central processor seconds.

The ensemble mean square pressures for the acoustic optimum solution, the feasible aerodynamic optimum and the non-optimum solution were found to be 0.1082×10^{-4}, 0.1166×10^{-4}, and 0.1132×10^{-4}, respectively. The corresponding root mean square of the acoustic pressures were 18.044 N/m^2, 18.752 N/m^2, and 18.456 N/m^2. The computed results are shown in Table 2.

To give a comparison of the order of magnitudes of the root mean square of acoustic pressures, we note that the root mean square of the pressure for the fundamental harmonic of the propeller with the same operating conditions, except slight difference in power coefficient is 15.4 N/m^2 obtained from Fig. 5 of Ref. 13. It should be noted that no far field or near field assumption is made in the present formulation while the root mean square pressures shown in Fig. 5 of Ref. 13 were obtained with the assumptions that the field point is in the far field and the radial integrals are replaced by an effective radius of the order of $0.8R_p$.

Table 2

Acoustic Optimum Solution for a Free-Running
Two-Bladed Propeller with Operating Conditions Given in Example 3.

$G_1 = 0.03651389$ $G_2 = -0.00863877$ $G_3 = -0.00565379$
$G_4 = 0.00087471$ $G_5 = 0.00018965$ $G_6 = -0.00171678$
$G_7 = -0.00000712$ $G_8 = 0.00059500$ $G_9 = -0.00010552$
$G_{10} = 0.00031216$

$C_{Ti} = 0.35560$
$C_{Pi} = 0.44604$
$\eta_i = 0.79724$

Ensemble mean square = 0.1082×10^{-4}

r	q	u_a^*	u_t^*	Γ
0.2	-1.00	0.08376	-0.00831	0.0
0.3	-0.75	0.09020	-0.08432	0.02606
0.4	-0.50	0.21581	-0.15674	0.03961
0.5	-0.25	0.28651	-0.18653	0.04639
0.6	0.00	0.23990	-0.12160	0.04225
0.7	0.25	0.15667	-0.06066	0.03277
0.8	0.50	0.11000	-0.03649	0.02328
0.9	0.75	0.07638	-0.01579	0.01268
1.0	1.00	-0.12592	0.03202	0.0

REFERENCES

1. Rankine, W. J. M., On the Mechanical Principles of the Action of Propeller. Trans. Inst. Nav. Arch., Vol. 6, pp. 13, 1865.

2. Froude, R. E., On the Part Played in Propulsion by Differences of Fluid Pressure. Trans. Inst. Nav. Arch., Vol. 30, pp. 390, 1889.

3. Froude, W., On the Elementary Relation between Pitch Slip, and Propulsive Efficiency. Trans. Inst. Nav. Arch., Vol. 19, pp. 47, 1878.

4. Drzewiecki, S., Theorie Generale de l'Helice. Paries, 1920.

5. Lanchester, F. W., Aerodynamics, Constable & Company, Ltd., London, 1907,

6. Goldstein, S., On the Vortex Theory of Screw Propellers Proceedings of the Royal Society (London), Series A, Vol. 63, pp. 440-465, 1929.

7. Ludwieg, H. and Ginzel, I., On the Theory of Screws with Wide Blades. Aerodynamische Versuchsenstalt, Goettingen, Report 44/A/08, 1944.

8. Lerbs, H. W., Moderately Loaded Propellers with a Finite Number of Blades and an Arbitrary Distribution of Circulation. Trans. The Society of Naval Architects and Marine Engineers (SNAME), Vol. 60, pp. 73-117, 1952.

9. Pien, P. C., The Calculation of Marine Propellers Based on Lifting-Surface Theory. J. of Ship Research, Vol. 5, No. 2, pp. 1-14, 1961.

10. Kerwin, J. E. and Leopold, R., A Design Theory for Subcavitating Propellers, Trans. SNAME, Vol. 72, pp. 294-335, 1964.

11. Morgan, Wm. B, Silovic, V. and Denny, S. B., Propeller Lifting-Surface Corrections. SNAME, Vol. 76, pp. 309-347, 1968.

12. Gutin, L., On the Sound Field of a Rotating Propeller. NACA TM 1195, 1948. (From Physik. Zeitscher. der Sojetunion, Bd 9, Heftl, pp. 57-71, (1936).

13. Garrick, I. E. and Watkins, C. E., A Theoretical Study of the Effect of Forward Speed on the Free-Space Sound Pressure Field Around Propellers. NACA Rep. 1198, pp. 961-976, 1954.

14. Lighthill, M. J., On Sound Generated Aerodynamically, I. General Theory. Proc. Roy. Soc. A221, pp. 564-587, 1952.

15. Lighthill, M. J., On Sound Generated Aerodynamically, II. Turbulence as a Source of Sound. Proc. Roy. Soc. A222, 1, 1954.

16. Ffowcs Williams, J. E. and Hawkings, D. L., Sound Generation by Turbulence and Surfaces in Arbitrary Motion. Philosophical Transactions of the Royal Society of London, Series A, 264, pp. 321-342, 1969.

17. Lowson, M. V., The Sound Field for Singularities in Motion. Proc. Roy. Soc. Series A 286, pp. 559-572, 1965.

18. Farassat, F., The Acoustic Far-Field of Rigid Bodies in Arbitrary Motion. J. of Sound and Vibration. 32(3), pp. 387-405, 1974.

19. Farassat, F., Some Research on Helicopter Rotor Noise Thickness and Rotational Noise. The Second Interagency Symposium on University Research in Transportation Noise North Carolina State University, Raleigh North Carolina, June 5-7, 1974.

20. Hawkings, D. L. and Lowson, M. V., Noise of High Speed Rotors. AIAA Second Aero-Acoustic Conference, Hampton, VA., March 24-26, 1975.

21. Hawkings, D. L. and Lowson, M. V., Theory of Open Supersonic Rotor Noise. J. of Sound and Vibration, 36(1), pp. 1-22, 1974.

22. Lowson, M. V., Theoretical Analysis of Compressor Noise. J. of the Acoustical Society of America, 47, pp. 371-385, 1970.

23. Goldstein, M., Aeroacoustics. National Aeronautical and Space Administration Washington, D.C., 1974.

24. AIAA Selected Reprint Series/Volume XI, Aerodynamic Noise. Edited by A. Goldburg, 1970.

25. Fuchs, H. V. and Michalke, A., Introduction to Aerodynamic Noise Theory. Progress in Aerospace Vol. 14, pp. 229-297, 1973.

26. Karamcheti, K. and Yu, Y. H., Aerodynamic Deisgn of a Rotor Blade for Minimum Noise Radiation. AIAA Paper No. 74-571, AIAA Seventh Fluid and Plasma Dynamics Conference Palo Alto, California/June 17-19, 1974.

27. Bisplinghoff, R. L., Ashley, H. and Halfman, R. L., Aeroelasticity. Addison-Wesley Publishing Company, Inc., 1957.

28. Wrench, J. W., The Calculation of Propeller Induction Factors. DTMB Report 1116, February 1957.

29. Box, M. J., A New Method of Constrained Optimization and a Comparison with other Methods. Comp. J. 8, pp. 42-52, 1965.

30. Basu, N. K., On Double Chebyshev Series Approximation. SIAM J. Numer. Anal. Vol. 10, No. 3, pp. 496-505, June 1973.

31. Richardson, J. A. and Kuester, J. L., The Complex Method for Constrained Optimization. Comm. ACM 16, pp. 487-489, Aug. 1973.

32. Luke, Y. L., The Special Functions and Their Applications. Vol. 1 and Vol. 2, Academic Press, New York and London, 1969.

33. Oberhettinger, F., Tabellen zur Fourier Transformation. Springer-Verlag, Berlin. Goettingen. Heidelberg. 1957.

34. Lebedev, N. N., Special Functions and Their Applications. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1965.

35. Cheng, H. M., Hydrodynamic Aspect of Propeller Design Based of Lifting Surface Theory: Part I - Uniform Chordwise Load Load Distribution. David Taylor Model Basin Report 1802, 1964.

APPENDIX A

EVALUATION OF INDUCTION FACTORS

Since we are only interested in the axial induction factor and the tangential induction factor, the radial induction factor will not be considered.

Integrating by parts, the tangential induction factor, Eq. (55), may be expressed in terms of the axial induction factor as

$$I_t(r,\rho) = -\frac{\lambda_i(\rho)}{r}\left\{ I_a(r,\rho) - \frac{B(\rho - r)}{\lambda_i(\rho)} \right\}$$

With this relationship, only the evaluation of the axial induction factor needs to be discussed. In addition to Wrench's modified formulas (28, 1957), an alternate method is presented.

I. Wrench's Modified Formulas

The Wrench's modified formulas for evaluation of the axial and tangential induction factors may be summarized as

$$I_a(r,\rho) = B y_0 \left(1 - \frac{y}{y_0}\right)\left(1 - 2B y_0 F_1\right) \qquad (r < \rho)$$

$$= 2B^2 y_0 y \left(1 - \frac{y_0}{y}\right) F_2 \qquad (r > \rho) \qquad \text{(A-1)}$$

$$I_t(r,\rho) = -2B^2 y_0 \left(1 - \frac{y_0}{y}\right) F_1 \qquad (r < \rho)$$

$$= -B\left(1 - \frac{y_0}{y}\right)\left(1 + 2B\, y_0 F_2\right) \qquad (r > \rho) \qquad \text{(A-2)}$$

where

$$y = \frac{r}{\lambda_i(\rho)}$$

$$y_0 = \frac{\rho}{\lambda_i(\rho)}$$

$$F_1 = -\frac{1}{2By_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{\frac{1}{4}}\left\{\frac{1}{u^{-1}-1} + \frac{1}{24B}\left(\frac{9y_0^2+2}{(1+y_0^2)^{3/2}}\right.\right.$$

$$\left.\left. + \frac{3y^2-2}{(1+y^2)^{3/2}}\right)\log\left(1 + \frac{1}{u^{-1}-1}\right)\right\} \qquad \text{(A-3)}$$

$$F_2 = \frac{1}{2By_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{\frac{1}{4}}\left\{\frac{1}{u-1} - \frac{1}{24B}\left(\frac{9y_0^2+2}{(1+y_0^2)^{3/2}}\right.\right.$$

$$\left.\left. + \frac{3y^2-2}{(1+y^2)^{3/2}}\right)\log\left(1 + \frac{1}{u-1}\right)\right\} \qquad \text{(A-4)}$$

$$u = \left(\frac{\sqrt{1+y^2}-1}{y}\,\frac{y_0}{\sqrt{1+y_0^2}-1}\right)^B e^{\left(\sqrt{1+y^2}-\sqrt{1-y_0^2}\right)B} \qquad \text{(A-5)}$$

For detailed derivation of Eqs. (A-1) and (A-2), the reader is referred to Ref. 28.

II. Alternate Method

For numerical computation, the integral of Eq. (54) is split up into four parts:

$$I_a(r,\rho) = I_a^1(r,\rho) + I_a^2(r,\rho) + I_a^3(r,\rho) + I_a^4(r,\rho) \tag{A-6}$$

where

$$I_a^1(r,\rho) = \int_{\epsilon}^{2\pi L} \frac{(\rho - r)\rho(\rho - r\cos\mu)}{\{\rho^2 + r^2 - 2\rho r\cos\mu + \lambda_i^2(\rho)\mu^2\}^{3/2}}\, d\mu \tag{A-7}$$

$$I_a^2(r,\rho) = \int_0^{2\pi L} \sum_{K=2}^{B} \frac{(\rho - r)\rho(\rho - r\cos(\mu + \delta_K))}{\{\rho^2 + r^2 - 2r\rho\cos(\mu + \delta_K) + \lambda_i^2(\rho)\mu^2\}^{3/2}}\, d\mu \tag{A-8}$$

$$I_a^3(r,\rho) = \int_0^{\epsilon} \frac{(\rho - r)\rho(\rho - r\cos\mu)}{\{\rho^2 + r^2 - 2\rho r\cos\mu + \lambda_i^2(\rho)\mu^2\}^{3/2}}\, d\mu \tag{A-9}$$

$$I_a^4(r,\rho) = \int_{2\pi L}^{\infty} \sum_{K=1}^{B} \frac{(\rho - r)\rho(\rho - r\cos(\mu + \delta_K))}{\{\rho^2 + r^2 - 2\rho r\cos(\mu + \delta_K) + \lambda_i^2(\rho)\mu^2\}^{3/2}}\, d\mu \tag{A-10}$$

where L is an integer, chosen such that

$$2\pi\, L\,\lambda_i(\rho) > 2$$

and $\quad 0 < \varepsilon << 1$

1. Evaluation of $I_a^1(r,\rho)$ and $I_a^2(r,\rho)$

To evaluate these integrals the interval of each integration is divided into several subintervals. A 25-point Legendre-Gauss formula is used in each subinterval.

2. Evaluation of $I_a^3(r,\rho)$

In this case, μ is small, and $\cos\mu$ may be substituted by $1 - \mu^2/2$. With this substitution Eq. (A-9) may be approximated as

$$I_a^3(r,\rho) = \int_0^{\epsilon} \frac{(\rho - r)\,\rho\,\{(\rho - r) + \frac{1}{2} r\mu^2\}}{\{(\rho - r)^2 + (r\rho + \lambda_i^2(\rho))\mu^2\}^{3/2}} \qquad \text{(A-11)}$$

Upon integrating Eq. (A-11), we have

$$I_a^3(r,\rho) = \frac{\rho\,\epsilon}{\sqrt{(\rho - r)^2 + (r\rho + \lambda_i^2(\rho))\,\epsilon^2}} + \frac{\frac{1}{2}\rho r(\rho - r)}{(r\rho + \lambda_i^2(\rho))} \Bigg\{ - \frac{\epsilon}{\sqrt{(\rho - r)^2 + (r\rho + \lambda_i^2(\rho))\epsilon^2}} + \frac{1}{\sqrt{r\rho + \lambda_i^2(\rho)}} \log \left(\frac{\epsilon\sqrt{r\rho + \lambda_i^2(\rho)} + \sqrt{(\rho - r)^2 + (r\rho + \lambda_i^2(\rho))\,\epsilon^2}}{|\rho - r|} \right) \Bigg\} \qquad \text{(A-12)}$$

3. Evaluation of $I_a^4(r,\rho)$

Since r and ρ are no greater than one and $2\pi L\lambda_i(\rho) > 2$, the integrand of Eq. (A-10) can be expanded as a hypergeometric series ${}_1F_0$ (32, 1969)

$$\sum_{K=1}^{B} \frac{(\rho-r)\{\rho-r\cos(\mu+\delta_K)\}}{\{\rho^2+r^2-2\rho r\cos(\mu+\delta_K)+\lambda_i^2(\rho)\mu^2\}^{3/2}}$$

$$=\sum_{K=1}^{B}(\rho-r)\rho\{\rho-r\cos(\mu+\delta_K)\}\left\{\frac{1}{\lambda_i^3(\rho)\mu^3}\right\}{}_1F_0\left(-\frac{3}{2};\frac{4(p+q\cos(\mu+\delta_K))}{\lambda_i^2(\rho)\mu^2}\right)$$

$$=(\rho-r)\sum_{n=0}^{\infty}A_n\sum_{i=0}^{n}C_i^n p^{n-i}q^i u\{\rho^2\,\mathrm{COSF}(u,2n+3,i)$$

$$-\rho r\,\mathrm{COSF}(u,2n+3,i+1)\} \qquad \text{(A-13)}$$

where

$$p = (r^2 + \rho^2)/4$$

$$q = -r\rho/2$$

$$A_n = \frac{(-1)^n\,\Gamma(n+\frac{3}{2})\left(\frac{2}{\lambda_i(\rho)u}\right)^{2n}}{\Gamma(\frac{3}{2})\,\Gamma(n+1)\,(\lambda_i(\rho)u)^3}$$

$$u = 2\pi L$$

$$\mathrm{COSF}(u,n,i) = \int_1^{\infty}\frac{\sum_{K=1}^{B}\cos^i(\delta_K+ut)}{t^n}\,dt \qquad \text{(A-14)}$$

Once COSF have been calculated, $I_a^4(r,\rho)$ can be evaluated by direct substitution. It is noted that COSF does not depend on $\lambda_i(\rho)$ but does depend on the number of propeller blades, B, and parameters u, n, and i.

Evaluation of COSF

The following are some useful formulas for evaluating COSF.

$$\sum_{K=1}^{B} \sin n \frac{2\pi}{B}(K-1) = 0 \quad \text{for any n and B} \tag{A-15}$$

$$\sum_{K=1}^{B} \cos n \frac{2\pi}{B}(K-1) = B \quad \text{if n/B is an integer} \tag{A-16}$$

$$= 0 \quad \text{if n/B is not an integer}$$

$$\cos^{2i}\theta = \frac{C_i^{2i}}{2^{2i}} + \frac{1}{2^{2i-1}} \sum_{j=1}^{i} C_{i-j}^{2i} \cos 2j\theta \tag{A-17}$$

$$\cos^{2i-1}\theta = \frac{1}{2^{2i-2}} \sum_{j=1}^{i} C_{i-j}^{2i-1} \cos(2j-1)\theta \tag{A-18}$$

Using these formulas with the sine-cosine product relations, we have

$$\sum_{K=1}^{B} \cos^{2i}(\delta_K + ut) = B\left(\frac{C_i^{2i}}{2^{2i}} + \frac{1}{2^{2i-1}} \sum_{j=1}^{i} C_{i-j}^{2i} \cos 2jut \ \delta_{2j/B}\right) \tag{A-19}$$

$$\sum_{K=1}^{B} \cos^{2i-1}(\delta_K + ut) = \frac{B}{2^{2i-2}}\left(\sum_{j=1}^{i} C_{i-j}^{2i-1} \cos(2j-1)ut \ \delta_{(2j-1)/B}\right. \tag{A-20}$$

where

$$\delta_x = 0 \quad \text{if x is not an integer}$$

$$= 1 \quad \text{if x is an integer}$$

Now, with Eqs. (A-19) and (A-20), we are able to express COSF in terms of summations of the generalized cosine integral, CI. Dividing Eqs. (A-19) and (A-20) by t^n and then integrating from 1 to infinity with respect to t, we have

$$COSF(u,n,2i) = \int_1^\infty \frac{\sum_{k=1}^{B} \cos^{2i}(\delta_k + ut)}{t^n}\, dt$$

$$= B\left\{\frac{C_i^{2i}}{2^{2i}}\frac{1}{n-1} + \frac{1}{2^{2i-1}}\sum_{j=1}^{i} C_{i-j}^{2i}\, CI(2ju,n)\,\delta_{2j/B}\right\} \quad \text{(A-21)}$$

$$COSF(u,n,2i-1) = \int_1^\infty \frac{\sum_{k=1}^{B} \cos^{2i-1}(\delta_k + ut)}{t^n}\, dt$$

$$= \frac{B}{2^{2i-2}} \sum_{j=1}^{i} C_{i-j}^{2i-1}\, CI\big((2j-1)u, n\big)\ \delta_{(2j-1)/B} \quad \text{(A-22)}$$

where CI is the generalized cosine integral defined as

$$CI(a,n) = \int_1^\infty \frac{\cos at}{t^n}\, dt \quad \text{(A-23)}$$

Asymptotic expansion of CI(a,n)

Applying integration by parts to Eq. (A-23), we obtain (33, 1957)

$$CI(a,2n) = \frac{(-1)^{n+1} a^{2n-1}}{(2n-1)!} \left\{ \frac{\cos a}{a} P_0^{n-1}(a) + \frac{\sin a}{a} Q_0^{n-2}(a) + si(a) \right\} \quad \text{(A-24}$$

$$CI(a,2n-1) = \frac{(-1)^n a^{2n-2}}{(2n-2)!} \left\{ \frac{\cos a}{a} Q_0^{n-2}(a) - \frac{\sin a}{a} P_0^{n-2}(a) + Ci(a) \right\} \quad \text{(A-25)}$$

where

$$si(a) = -\int_a^\infty \frac{\sin t}{t} dt = -\frac{\pi}{2} + Si(a) = -\frac{\pi}{2} + \int_0^a \frac{\sin t}{t} dt \quad \text{(A-26)}$$

$$Si(a) = \int_0^a \frac{\sin t}{t} dt \quad \text{(A-27)}$$

$$Ci(a) = \int_\infty^a \frac{\cos t}{t} dt \quad \text{(A-28)}$$

$$P_i^j(a) = \sum_{k=i}^{j} \frac{(-1)^k (2k)!}{a^{2k}} \quad \text{(A-29)}$$

$$Q_i^j(a) = \sum_{k=i}^{j} \frac{(-1)^k (2k+1)!}{a^{2k+1}} \quad \text{(A-30)}$$

The asymptotic representations of Ci(a) and $\frac{\pi}{2}$ - Si(a) may be found in those books which deal with special functions. The following are taken from Ref. 34, with some change in notation

$$Ci(a) = \frac{\sin a}{a} P_0^\infty(a) - \frac{\cos a}{a} Q_0^\infty(a) \quad \text{(A-31)}$$

$$\frac{\pi}{2} - Si(a) = \frac{\cos a}{a} P_0^\infty(a) + \frac{\sin a}{a} Q_0^\infty(a) \quad \text{(A-32)}$$

where

$$P_0^{\infty}(a) = \sum_{k=0}^{n} \frac{(-1)^k (2k)!}{a^{2k}} + O(|a|^{-2n-2}) \tag{A-33}$$

$$Q_0^{\infty}(a) = \sum_{k=0}^{n} \frac{(-1)^k (2k+1)!}{a^{2k+1}} + O(|a|^{-2n-3}) \tag{A-34}$$

Using these expansions in Eqs. (A-24) and (A-25), we obtain

$$CI(a,2n) = \frac{(-1)^n a^{2n-1}}{(2n-1)!} \left\{ \frac{\cos a}{a} P_n(a) + \frac{\sin a}{a} Q_{n-1}(a) \right\} \tag{A-35}$$

$$CI(a,2n-1) = \frac{(-1)^n a^{2n-2}}{(2n-2)!} \left\{ \frac{\cos a}{a} Q_{n-1}(a) - \frac{\sin a}{a} P_{n-1}(a) \right\} \tag{A-36}$$

where

$$P_n(a) = P_n^{\infty}(a)$$

$$Q_n(a) = Q_n^{\infty}(a)$$

For convenience of numerical computation, we define

$$R_n^*(a) = \frac{1}{a} \left\{ 1 - \frac{(2n)(2n+1)}{a^2} \left(1 - \frac{(2n+2)(2n+3)}{a^2} \left(1 - \frac{(2n+4)(2n+5)}{a^2} + \cdots \right) \right) \right\} \tag{A-37}$$

$$S_n^*(a) = \frac{2n}{a^2} \left\{ 1 - \frac{(2n+1)(2n+2)}{a^2} \left(1 - \frac{(2n+3)(2n+4)}{a^2} \left(1 - \frac{(2n+5)(2n+6)}{a^2} + \cdots \right) \right) \right\} \tag{A-38}$$

$$T_n^*(a) = \frac{1}{a}\left\{1 - \frac{(2n-1)2n}{a^2}\left(1 - \frac{(2n+1)(2n+2)}{a^2}\left(1 - \frac{(2n+3)(2n+4)}{a^2} + \cdots\right)\right)\right\} \tag{A-39}$$

$$U_n^*(a) = \frac{2n-1}{a^2}\left\{1 - \frac{2n(2n+1)}{a^2}\left(1 - \frac{(2n+2)(2n+3)}{a^2}\left(1 - \frac{(2n+4)(2n+5)}{a^2} + \cdots\right)\right)\right\} \tag{A-40}$$

Then,

$$CI(a,2n) = S_n^*(a)\cos a - R_n^*(a)\sin a \tag{A-41}$$

$$CI(a,2n-1) = U_n^*(a)\cos a - T_n^*(a)\sin a \tag{A-42}$$

With a proper choice of L and ε, this alternate method will yield induction factors to any desired accuracy. It is noted that this method can be extended to the evaluation of both the tangential and radial induction factors.

APPENDIX B

EVALUATION OF COEFFICIENTS h^a_{ij} AND h^t_{ij}

To evaluate the coefficients h^a_{ij} and h^t_{ij}, we give two backward recurrence formulas for evaluating polynomials in Chebyshev form

1. $$P(q) = \sum_{j=0}^{n}{}' A_j T_j(q) = \frac{B_0 - B_2}{2}$$

where

$$B_{n+1} = B_{n+2} = 0$$

$$B_K = 2q\, B_{K+1} - B_{K+2} + A_K \quad K = n, n-1, \cdots, 0 \tag{B-1}$$

2. $$P(q) = \sum_{j=0}^{m} C_{2j+1} T_{2j+1}(q) = q\,(B_0 - B_1)$$

where

$$B_{m+1} = B_{m+2} = 0$$

$$B_K = 2t\, B_{K+1} - B_{K+2} + C_{2K+1} \qquad K = m, m-1, \cdots, 0 \tag{B-2}$$

$$t = 2q^2 - 1$$

To evaluate the coefficients h^a_{ij}, we put

$$C_{rs} = I_a\left(q_r^{(n+1)}, q_{os}^{(n+1)}\right) \tag{B-3}$$

and

$$P_{rj} = \sum_{s=0}^{n} C_{rs} T_j\left(q_{os}^{(n+1)}\right) \tag{B-4}$$

so that Eq. (119) becomes

$$h^a_{ij} = \frac{4}{(n+1)^2} \sum_{r=0}^{n} P_{rj} T_i\left(q_r^{(n+1)}\right) \tag{B-5}$$

Define

$$x_j^{(n+1)} = \cos \frac{j\pi}{2(n+1)}$$

Then,

$$T_j(q_{os}^{(n+1)}) = T_{2s+1}(x_j^{(n+1)}) \tag{B-6}$$

and

$$T_i(q_r^{(n+1)}) = T_{2r+1}(x_i^{(n+1)}) \tag{B-7}$$

Upon substituting Eqs. (B-6) and (B-7) into Eqs. (B-4) and (B-5), respectively, we have

$$P_{rj} = \sum_{s=0}^{n} C_{rs} T_{2s+1}(x_j^{(n+1)}) \tag{B-8}$$

$$h_{ij}^{a} = \frac{4}{(n+1)^2} \sum_{r=0}^{n} P_{rj} T_{2r+1}(x_i^{(n+1)}) \tag{B-9}$$

The advantage of Eqs. (B-8) and (B-9) is that P_{rj} and h_{ij}^{a} can be evaluated by using the backward recurrence formula (B-2)

The coefficients h_{ij}^{t} are obtained by replacing $I_a(q_r^{(n+1)}, q_{os}^{(n+1)})$ in Eq. (B-3) by $I_t(q_r^{(n+1)}, q_{os}^{(n+1)})$.

APPENDIX C

EVALUATION OF RETARDED DISTANCE R

Let

$$f(R) = \frac{R}{X} - \frac{1}{1-M_F^2}\Big\{ -M_F \cos \Theta + \sqrt{M_F^2 \cos^2 \Theta + (1-M_F^2)\left(1+\frac{\rho^2}{X} - 2\,\frac{\rho}{X}\sin\Theta \cos(\phi_o+\delta_K+M_P R)\right)}\Big\} \tag{C-1}$$

$$f'(R) = \frac{1}{X} - \frac{M_P \frac{\rho}{X} \sin\Theta \sin(\phi_o + \delta_K + M_P R)}{\sqrt{M_F^2 \cos^2\Theta + (1-M_F^2)\left(1+\frac{\rho^2}{X} - 2\,\frac{\rho}{X}\sin\Theta \cos(\phi_o+\delta_K+M_P R)\right)}} \tag{C-2}$$

$$R_0 = \frac{X}{1-M_F^2}\left\{ -M_F \cos\Theta + \sqrt{1 - M_F^2 \sin^2\Theta} \right\}$$

Then, R is found by the Newton method:

$$R_n = R_{n-1} - \frac{f(R_{n-1})}{f'(R_{n-1})} \tag{C-4}$$

APPENDIX D

FORTRAN LISTING OF COMPUTER PROGRAM

```
      PROGRAM PROPEL(INPUT,OUTPUT,COSGRA,PKOATD,
     $ TAPE1=COSGRA,TAPE2=PKOATD,TAPE6=OUTPUT)
C
C     PURPOSE
C
C     TO SOLVE PROBLEMS I THROUGH IV LISTED BELOW
C     BASED ON THE TECHNIQUE DEVELOPED IN PART 2.
C
C       DEFINITIONS:
C
C           VP=REFERENCE VELOCITY(USUALLY,
C              VP= ADVANCE SPEED OF PROPELLER)
C       NBLADE=NUMBER OF PROPELLER BLADES
C           RP=REFERENCE LENGTH (RADIUS OF PROPELLER)
C       T(N,Q)=CHEBYSHEV POLYNOMIAL OF THE FIRST KIND
C              OF DEGREE N (-1.GE. Q .LE.1)
C       U(N,Q)=CHEBYSHEV POLYNOMIAL OF THE SECOND KIND
C              OF DEGREE N
C           RH=HUB RADIUS OF PROPELLER
C           RC=RADIAL COORDINATE OF BLADE SECTION
C              RH .GT. RC .LE. RP
C        OMEGA=ANGULAR VELOCITY OF PROPELLER
C       RAMDAI=HYDRODYNAMIC ADVANCE COEFFICIENT
C            Q=(2*R-HUB-1)/(1-HUB)  (-1.GE. Q .LE.1)
C              WHERE HUB=RH/RP, HUB .GE. R=RC/RP .LE. 1)
C            T=TOTAL THRUST
C            P=TOTAL POWER
C          MSP=ENSEMBLE MEAN SQUARE OF ACOUSTIC PRESSURE
C       RAMDAP=REFERENCE ADVANCE COEFFICIENT
C              (RAMDAP=VP/(RP*OMEGA))
C           VF=ADVANCE SPEED OF PROPELLER/VP
C           FM=ADVANCE MACH NUMBER OF PROPELLER
```

```
C          TM=TIP MACH NUMBER
C          CT=THRUST COEFFICIENT=T/(0.5*DENSITY*
C             VP**2*PAI*RP**2)
C          CP=POWER COEFFICIENT=P/(0.5*DENSITY*
C             VP**3*PAI*RP**2)
C        ETAI=IDEAL EFFICIENCY=CT*VF/CP
C      ENSMSQ=NON-DIMENSIONAL ENSEMBLE MEAN SQUARE
C             OF THE ACOUSTIC PRESSURE
C            =MSP/(DENSITY*VP**2/(4*PAI))
C      XDISTA=NON-DIMENSIONAL DISTANCE BETWEEN THE
C             OBSERVER AND THE CENTER OF PROPELLER
C      AZIMUT=AZIMUTH ANGLE OF OBSERVER(IN DEGREE)
C             (0 .GE. AZIMUT .LE. 90(DEGREE) WHEN
C             THE OBSERVER IS BEHIND THE PROPELLER
C             DISK
C          UA=NON-DIMENSIONAL AXIAL COMPONENT OF
C             INDUCED VELOCITY=AXIAL INDUCED VELOCITY/VP
C          UT=NON-DIMENSIONAL TANGENTIAL COMPONENT OF
C             INDUCED VELOCITY=TANGENTIAL INDUCED
C             VELOCITY/VP
C       GAMMA=NON-DIMENSIONAL CIRCULATION=CIRCULATION/
C             (2.*PAI*VP*RP)
C            =SQRT(1-Q**2)*SUMMATIONG(I)*U(I-1,Q),
C             I=1, 2, 3....
C
C     REMARK:  1. ALL INPUT AND OUTPUT ARE IN NON-DIMENSIONAL
C                 FORMS
C              2. INPUT DATA ARE MADE IN
C                A. MAIN PROGRAM
C                B. SUBROUTINE COMPLEX
C                C. SUBROUTINE JCNST1
C                D. SUBROUTINE NOPTIML
C                E. SUBROUTINE INDA12
C
C             3. THIS PROGRAM CONSISTS OF A MAIN PROGRAM AND
C                49 SUBPROGRAMS:
```

```
C                                1. SUBROUTINE COMPLEX,
C            2. SUBROUTINE MEANSQF, 3. SUBROUTINE JFUNC,
C            4. SUBROUTINE AECOEF, 5. SUBROUTINE JCNST1,
C            6. SUBROUTINE JCONSX, 7. SUBROUTINE JCEK1,
C            8. SUBROUTINE JCENT, 9. SUBROUTINE AERODYN,
C            10. SUBROUTINE NOPTIML, 11. SUBROUTINE CTP12,
C            12. SUBROUTINE AEROCF, 13. FUNCTION RAIJKM,
C            14. SUBROUTINE CIRCU, 15. SUBROUTINE UATCHBY,
C            16. SUBROUTINE HATIJ, 17. FUNCTION RAMDAF,
C            18. SUBROUTINE RAMCOE, 19. SUBROUTINE DOUCHB,
C            20. SUBROUTINE CHEBCF, 21. SUBROUTINE SUMODD,
C            22. SUBROUTINE SUMCHB, 23. SUBROUTINE ZEROS,
C            24. SUBROUTINE DOUSUM, 25. SUBROUTINE FACTOR,
C            26. FUNCTION CIRCLF, 27. SUBROUTINE INDVEL,
C            28. SUBROUTINE VELOCIT, 29. SUBROUTINE TCHEBY,
C            30. SUBROUTINE UCHEBY, 31. SUBROUTINE INDFACT,
C            32. SUBROUTINE INDA12, 33. FUNCTION GQUZ25,
C            34. SUBROUTINE GQUDAD, 35. SUBROUTINE INDA3,
C            36. FUNCTION AIND3, 37. SUBROUTINE INDA4,
C            38. FUNCTION AIND4, 39. SUBROUTINE COEAN,
C            40. FUNCTION FAXIAL, 41. FUNCTION DLGAMA,
C            42. SUBROUTINE WRENF, 43. SUBROUTINE MEANSQ,
C            44. SUBROUTINE PRESUF, 45. SUBROUTINE FKOAT,
C            46. SUBROUTINE RETARD, 47. SUBROUTINE PKOAT,
C            48. FUNCTION FEWTON, 49. SUBROUTINE ABSIAS.
C
C
C  PROBLEM I:
C
C      GIVEN: NBLADE, HUB
C             VF
C             RAMDAP( OR FM AND TM)
C             RAMDAI
C
C      DETERMINE: AERODYNAMIC OPTIMUM CIRCULATION
C                 DISTRIBUTION
```

```
C                  INDUCED VELOCITY COMPONENTS
C                  CT, CP, AND IDEAL EFFICIENCY
C                  ENSMSQ(OPTIONAL)
C
C  PROBLEM II:
C
C     GIVEN: NBLADE, HUB
C            VF
C            RAMDAP (OR FM AND TM)
C            CT (OR CP)
C
C     DETERMINE: AERODYNAMIC OPTIMUM CIRCULATION
C                DISTRIBUTION
C                INDUCED VELOCITY COMPONENTS
C                CP (OR CT) AND IDEAL EFFICIENCY
C                ENSMSQ(OPTIONAL)
C
C  PROBLEM III:
C
C     GIVEN: NBLADE, HUB
C            VF
C            RAMDAP (OR FM AND TM)
C            CT (OR CP)
C            THE TYPE OF CIRCULATION FUNCTION
C
C     DETERMINE: NON-OPTIMUM CIRCULATION DISTRIBUTION
C                INDUCED VELOCITY COMPONENTS
C                CP (OR CT) AND IDEAL EFFICIENCY
C                ENSMSQ(OPTIONAL)
C
C  PROBLEM IV:
C
C     GIVEN: NBLADE, HUB
C           FM, TM, AND VF
C           HUB
C           UPPER BOUND AND LOWER BOUND OF CT
```

```
C           UPPER BOUND AND LOWER BOUND OF CP
C           UPPER BOUND AND LOWER BOUND OF RAMDAI
C
C     DETERMINE: AEROACOUSTIC OPTIMUM CIRCULATION
C                DISTRIBUTION
C                INDUCED VELOCITY COMPONENTS
C                CT, CP, AND ETAI
C                MINIMUM ENSEMBLE MEAN SQUARE OF
C                ACOUSTIC PRESSURE
C
C
C
C  USAGE:
C
C  1. PROBLEM I:
C
C             A. ENSMSQ IS NOT DESIRED
C
C                INPUT: NBLADE,HUB
C                       VF, RAMDAP
C                       NTRY=1
C                       TRYRAM(1)=RAMDAI
C                       IPRBLM=1
C                       IMEAN=0
C
C            B. ENSMSQ IS DESIRED
C
C               INPUT: NBLADE, HUB
C                      VF, FM, TM (RAMDAP=FM/(TM*VF)
C                      XDISTA, AZIMUT(IN DEGREE)
C                      NTRY=1
C                      TRYRAM(1)=RAMDAI
C                      IPRBLM=1
C                      IMEAN=1
C
C  2. PROBLEM II:
```

```
C
C              A. ENSMSQ IS NOT DESIRED
C
C                 INPUT: NBLADE, HUB
C                        VF,RAMDAP (OR FM AND TM)
C                        CT(OR CP)
C                        IPRBLM=2
C                        IMEAN=0
C                        NTRY=NUMBER OF PRESET VALUES OF RAMDAI
C                        TRYRAM(I)=RAMDAI, I=1(1)NTRY
C                        CTP=CT IF CT IS SPECIFIED
C                        CTP=CP IF CP IS SPECIFIED
C                        NTP=1 IF CTP=CT
C                        NTP.NE.1 IF CTP=CP
C
C              B. ENSMSQ IS DESIRED
C
C                 INPUT:  NBLADE, HUB
C                         VF,FM,TM
C                         XDISTA,AZIMUT(IN DEGREE)
C                         NTRY=NUMBER OF PRESET VALUES
C                              OF RAMDAI
C                         TRYRAM(I)=RAMDAI,I=1(1)NTRY
C                         CTP=CT IF CT IS SPECIFIED
C                         CTP=CP IF CP IS SPECIFIED
C                         NTP=1 IF CTP=CT
C                         NTP.NE.1 IF CTP=CP
C                         IPRBLM=2
C                         IMEAN=1
C  3. PROBLEM III:
C
C              A. ENSMSQ IS NOT DESIRED
C
C                 INPUT: NBLADE, HUB
C                 VF, RAMDAP (OR FM AND TM)
C                 CTP=CT IF CT IS SPECIFIED
```

```
C                    CTP=CP IF CP IS SPECIFIED
C                    NTP=1 IF CTP=CT
C                    NTP .NE. 1 IF CTP = CP
C                    IPRBLM=3
C                    IMEAN=0
C                    GG(I), I=1,2,3,....
C                    (SEE SUBROUTINE NOPTIML)
C
C                    WHERE
C
C                     GAMMA=K * SQRT(1.-Q**2) *
C                     SUMMATION GG(I)*U(I-1,Q), I=1,2,....
C                    K: TO BE DETERMINED
C
C                B. ENSMSQ IS DESIRED
C
C                   INPUT: NBLADE, HUB
C                          VF, FM, TM
C                          XDISTA, AZIMUT
C                          CTP=CP IF CT IS SPECIFIED
C                          CTP=CP IF CP IS SPECIFIED
C                          NTP=1 IF CTP=CT
C                          NTP .NE.1 IF CTP=CP
C                          IPRBLM=3
C                          IMEAN=1
C                          GG(I), I=1, 2, 3, ....
C                          (SEE SUBROUTINE NOPTIML)
C                          WHERE
C
C                          GAMMA=K * SQRT(1.-Q**2) *
C                                SUMMATION GG(I)*U(I-1,Q)
C                                I=1, 2, 3, .........
C
C                          K: TO BE DETERMINED
C
C  4. PROBLEM IV.
```

```
C
C               INPUT: NBLADE,HUB
C                      VF, FM, TM
C                      XDISTA,AZIMUT
C                      UPPER BOUND AND LOWER BOUND OF CT
C                      UPPER BOUND AND LOWER BOUND OF CP
C                      UPPER BOUND AND LOWER BOUND OF RAMDAI
C                      AT THE ZEROS OF T(JRAMDA+1,Q)
C                      (FOR DETAIL, SEE SUBROUTINE COMPLEX)
C                      CTP=CT (OR CP) USED FOR FINDING
C                      THE FIRST APPROXIMATON,
C                      (CTP MUST SATISFY CONSTRAIN)
C                      NTP=1 IF CTP=CT
C                      NTP .NE. 1 IF CTP=CP
C                      NTRY=NUMBER OF ASSUMED VALUES OF RAMDAI
C                      TRYRAM(I)=RAMDAI I=1(1)NTRY
C                      IPRBLM=4
C                      IMEAN=1
C
C
C     PRECISION: SINGLE PRECISION
C
C     REQUIRED DATA FILES
C
C       COSGRA      (SEE PROGRAM COECOS) NEEDED ONLY WHEN
C                   THE ALTERNATE METHOD DEVELOPED IN APPENDIX A
C                   IS APPLIED TO COMPUTE THE INDUCTION FACTORS
C       PK0ATD      CONTAINS THE COEFFICIENTS, PK0, PKA,
C                   AND PKT, OF THE CHEBYSHEV EXPANSIONS OF
C                   THE KERNELS OF ACOUSTIC PRESURE
C                   NEEDED ONLY WHEN NTAPE .EQ. 2
C                   COMPUTED BY PROGRAM ITSELF IF NTAPE.NE.2
C
C     DESCRIPTION OF PARAMETERS
C
C       HUB           HUB RADIUS OF PROPELLER(INPUT)
```

```
C       NBLADE          NUMBER OF PROPELLER BLADES (INPUT)
C       TM              TIP MUCH NUMBER OF PROPELLER (INPUT)
C       FM              FORWARD MUCH NUMBER (INPUT)
C       VF              NON-DIMENSIONAL FORWARD VELOCITY OF
C                       PROPELLER W.R.T REFERENCE VELOCITY
C                       VP, IF VP IS CHOSEN TO BE THE FORWARD
C                       VELOCITY OF PROPELLER, VF=1 (INPUT)
C       CT              THRUST COEFFICIENT
C       CP              POWER COEFFICIENT
C       CTP             SPECIFIED CT OR CP (INPUT)
C       ETAI            IDEAL EFFICIENCY
C       JRAMDA          JRAMDA+1: NUMBER OF THE FUNCTION VALUES
C                       OF RAMDAI AT THE ZEROS OF T(JRAMDA+1,Q)
C                       THAT IS, THE NUMBER OF EXPLICIT
C                       INDENPENDENT VARIABLES (INPUT)
C       MCIRCU          NUMBER OF TERMS IN CHEBYSHEV EXPANSION OF
C                       THE CIRCULATION (INPUT)
C       ORAMDA(I)       RAMDAI(Q) AT ZEROS OF T(JRAMDA+1,Q)
C                       I=1(1)JRAMDA+1
C       OQZERO(I)       ZEROS OF T(JRAMDA+1,Q) THAT IS,
C
C                       OQZERO(I)=COS((2*I+1)*PAI/(2*(JRAMDA+1))
C
C       QCIRCU(I)       ZEROS OF T(MCIRCU,Q)
C                       USED IN CALCULATION OF THE COEFFICIENTS
C                       OF THE CHEBYSHEV EXPANSION OF CIRCULATION
C       XDISTA          NON-DIMENSIONAL DISTANCE BETWEEN THE
C                       PROPELLER AND THE OBSERVER (INPUT)
C       AZIMUT          AZIMUTH ANGLE OF THE OBSERVER IN DEGREE
C                       (INPUT)
C       IDEG            DEGREE OF THE DOUBLE CHEBYSHEV EXPANSION
C                       OF BOTH AXIAL AND TANGENTIAL INDUCTION
C                       FACTORS (INPUT)
C       ICHEBY          ICHEBY+1:  NUMBER OF POINTS USED IN THE
C                       COMPUTATION OF THE COEFFICIENTS OF THE
C                       DOUBLE CHEBYSHEV EXPANSION FOR INDUCTION
```

```
C                    FACTORS(INPUT)
C      QIND(I)       ZEROS OF T(ICHEBY+1,Q), THAT IS
C
C                    QIND(I)=COS((2*I+1)*PAI/(2*(ICHEBY+1))
C                    I=0(1)ICHEBY
C      EPSIND        LOWER LIMIT OF IA12 AND THE UPPER LIMIT
C                    OF IA3 (INPUT)
C      LUPIND        2*PAI*LUPIND: UPPER LIMIT OF IA12 AND
C                    LOWER LIMIT OF IA4 (INPUT)
C      KWENH         CONTROL PARAMETER
C                    IF KWENH=1, THE PROPELLER INDUCTION
C                    FACTORS ARE COMPUTED BY WRENCH'S
C                    FORMULA
C                    IF KWENH .NE.1 THE PROPELLER INDUCTION
C                    FACTORS ARE COMPUTED BY ALTERNATE METHOD
C                    (INPUT)
C      NNOISE        DEGREE OF THE CHEBYSHEV EXPANSIONS
C                    OF K0, KA, AND KT (INPUT)
C      NOSCHB        NOSCHB+1: NUMBER OF POINTS USED IN THE
C                    EVALUATION OF THE COEFFICIENTS OF THE
C                    CHEBYSHEV EXPANSIONS OF K0,KA, AND KT
C                    (INPUT)
C      QNOISE(I)     ZEROS OF T(NOSCHB+1,Q)
C      ETA(I)        ABSCISSAS OF THE 25-POINT GAUSS-LEGENDRE
C                    FORMULA
C      SQMEAN        MEAN SQUARE OF ACOUSTIC PRESSURE
C      IMEAN         PROBLEM CONTROL PARAMETER (INPUT)
C      IPRBLM        PROBLEM CONTROL PARAMETER (INPUT)
C      NZERO         LENGTH OF ARRAY ZERO (INPUT)
C      NOR           LENGTH OF ARRAY ORAMDA  (INPUT)
C      NTAPE         INPUT CONTROL PAREMETER (INPUT)
C                    NTAPE=0 IF DATA FILE PK0ATD IS NOT
C                    PROVIDED
C                    NTAPE=2 IF DATA FILE PK0ATD IS PROVIDED
C                    BY USER
C      ENS(I)        ENS(1)=ENSMSQ CORRESPONDING TO
```

```
C                              AERODYNAMIC OPTIMUM PROPELLER
C                       ENS(2)=ENSMSQ CORRESPONDING TO
C                              NON-OPTIMUM PROPELLER
C     GT(I)             GT(1)=COMPUTED CT CORRESPONDING
C                             AERODYNAMIC OPTIMUM PROPELLER
C                       GT(2)=COMPUTED CT CORRESPONDING TO
C                             NON-OPTIMUM PROPELLER
C     GP(I)             GP(1)=COMPUTED CP CORRESPONDING TO
C                             AERODYNAMIC OPTIMUM PROPELLER
C                       GP(2)=COMPUTED CP CORRESPONDING TO
C                             NON-OPTIMUM PROPELLER
C     CTRAIN            MIN(GT(I))
C     CPRAIN            MAX(GP(I))
C
C
C
      DIMENSION ZERO(51),ORAMDA(15),TRYRAM(5)
      COMMON /CONST/PAI
      COMMON /DATA1/JRAMDA
      COMMON /DATA2/A0RAMD,ARAMDA(14)
      COMMON /DATA3/IDEG,ICHEBY
      COMMON /DATA4/HA(31,31),HT(31,31)
      COMMON /DATA5/QIND(51)
      COMMON /DATA6/HUB
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /DATA9/EPSIND
      COMMON /DATA10/QCIRCU(15)
      COMMON /DATA11/MCIRCU
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA14/G(15)
      COMMON /DATA15/CT,CP,ETAI
      COMMON /DATA16/OQZERO(15)
      COMMON /DATA17/XDISTA,AZIMUT,ANGLE,SANGLE,CANGLE
      COMMON /DATA19/QNOISE(51)
      COMMON /DATA20/NNOISE,NOSCHB
      COMMON /DATA21/ETA(25)
```

```
      COMMON /DATA22/PKO(25,31),PKA(25,31),PKT(25,31)
      COMMON /DATA24/SQMEAN
      COMMON /DATA31/IVELDG,IVELCHB
      COMMON /DATA32/QVEL(51)
      COMMON /DATA33/UACHB(51),UTCHB(51)
      COMMON /DATA34/CTRAIN,CPRAIN
      COMMON /DATA35/CT1,CT2,CP1,CP2
      COMMON /DATA36/IMEAN,IPRBLM
      COMMON /DATA37/ENS(2)
      COMMON /DATA38/XX(2,15)
      COMMON /DATA39/GT(2),GP(2)
      PAI=3.141592653589793
C
C***  INPUT DATA
C
C
C     PROPELLER DATA:
C
      NBLADE=2
      HUB=0.2
C
C  AERODYNAMIC DATA:
C
      VF=1.
      FM=0.2
      TM=0.7426
      RAMDAP=FM/(TM*VF)
C
C  ACOUSTIC DATA:
C
      XDISTA=4.272
      AZIMUT=69.44
      ANGLE=PAI*AZIMUT/180.
      SANGLE=SIN(ANGLE)
      CANGLE=COS(ANGLE)
C
```

```
C     PROBLEM CONTROL PARAMETER
C
      IMEAN=1
      IPRBLM=4
C
C     WRITE OUT THE PROPELLER DATA
C
   60 FORMAT(///,5X,"******************************")
   61 FORMAT(5X,"*****                    *****")
   62 FORMAT(5X,"***** DEFINITIONS OF     *****")
   63 FORMAT(5X,"***** SYMBOLS            *****")
   64 FORMAT(5X,"******************************"//)
      WRITE(6,60)
      WRITE(6,61)
      WRITE(6,62)
      WRITE(6,63)
      WRITE(6,61)
      WRITE(6,64)
      WRITE(6,65)
   65 FORMAT(5X,"NBLADE=NUMBER OF PROPELLER BLADES"/,5X,
     $ "HUB=NON-DIMENSIONAL HUB RADIUS"/,5X,
     $ "VF=NON-DIMENSIONAL ADVANCE SPEED OF PROPELLER"/,5X,
     $ "MF=ADVANCE MACH NUMBER OF PROPELLER"/,5X,
     $ "MT=TIP MACH NUMBER OF PROPELLER"/,5X,
     $ "RAMDAP=MF/(MT*VF),REFERRENCE ADVANCE COEFFICIENT"/,5X,
     $ "XDISTA=DISTANCE OF OBSERVER FROM THE PROPELLER"/
     $ ,5X,"R=RADIAL COORDINATE OF PROPELLER BLADE SECTION"/,
     $ 5X,"Q=(2*R-HUB-1)/(1-HUB), -1 .GE. Q .LE. 1"/,5X,
     $ "CT=THRUST COEFFICIENT"/,5X,
     $ "CP=POWER COEFFICIENT"/,5X,
     $ "G(I)=COEFFICIENTS OF THE CHEBYSHEV EXPANSION"/,5X,
     $ "OF CIRCULATION, I.E. GAMMA"/,5X,
     $ "UA=AXIAL COMPONENT OF INDUCED VELOCITY"/,5X,
     $ "UT=TANGENTIAL COMPONENT OF INDUCED VELOCITY"/,5X,
     $ "RAMDAI=HYDRODYNAMIC ADVANCE COEFFICIENT"/,5X,
     $ "GAMMA=NON-DIMENSIONAL CIRCULATION"///)
```

```
50   FORMAT(///,5X,"*************************************")
51   FORMAT(5X,"*****                          *****")
52   FORMAT(5X,"***** PROPELLER DATA AND       *****")
53   FORMAT(5X,"***** PROGRAM CONTROL PARAMETERS  *****")
54   FORMAT(5X,"*************************************"//)
     WRITE(6,50)
     WRITE(6,51)
     WRITE(6,52)
     WRITE(6,53)
     WRITE(6,51)
     WRITE(6,54)
     WRITE(6,19) NBLADE,HUB,VF,FM,TM,RAMDAP,
    $ XDISTA,AZIMUT
19   FORMAT(//,5X,"NBLADE=",I2/,5X,"HUB=",F10.5/5X,
    $ "VF=",F10.5/5X,"MF=",F10.5/5X,"MT=",F10.5/5X,
    $ "RAMDAP=",F10.5/,5X,"XDISTA=",F10.5/5X,
    $ "AZIMUTH ANGLE=",F10.5,2X,"(DEGREE)"//)
C
C
C     PARAMETERS USED IN THE COMPUTATION OF
C     INDUCTION FACTOR
C
      KWENH=1
      EPSIND=0.
      LUPIND=4
C
C
C     PARAMETERS FOR NUMERCIAL SOLUTION OF
C     OPTIMAL CIRCULATION
C
C
      TRYRAM(1)=0.315
      TRYRAM(2)=0.32
      TRYRAM(3)=0.325
      NTRY=3
      NTP=1
```

```
      CTP=0.3556
      NRAM=5
      JRAMDA=9
      IDEG=10
      ICHEBY=10
      MCIRCU=10
      NNOISE=10
      NOSCHB=10
      IVELDG=2*IDEG+MCIRCU-1
      IVELCHB=IVELDG
C
C     PRINT OUT THE PROGRAM CONTROL PARAMETERS
C
      WRITE(6,20)JRAMDA,IDEG,ICHEBY,MCIRCU,
     $ NNOISE,NOSCHB
   20 FORMAT(5X,"JRAMDA=",I3/5X,"IDEG=",I3/5X,
     $ "ICHEBY=",I3/5X,"MCIRCU=",I3/5X,
     $ "NNOISE =",I3/5X,"NOSCHB=",I3//)
C
C
C     PROGRAM PARAMETERS
C
      NTAPE=2
      NZERO=51
      NOR=15
C
C***  END OF INPUT
C
C     COMPUTE QIND(I) THE ZEROS OF T(ICHEBY+1,Q)
C
      NDEGRE=ICHEBY+1
      CALL ZEROS(NDEGRE,NZERO,ZERO)
      DO 1 I=1,NDEGRE
      QIND(I)=ZERO(I)
    1 CONTINUE
C
```

```
C     COMPUTE QCIRCU(I), THE ZEROS OF T(MCIRCU,Q)
C
      NDEGRE=MCIRCU
      CALL ZEROS(NDEGRE,NZERO,ZERO)
      DO 2 I=1,NDEGRE
      QCIRCU(I)=ZERO(I)
    2 CONTINUE
C
C     COMPUTE OQZERO(I), THE ZEROS OF T(JRAMDA+1,Q)
      NDEGRE=JRAMDA+1
      CALL ZEROS(NDEGRE,NZERO,ZERO)
      DO 3 I=1,NDEGRE
      OQZERO(I)=ZERO(I)
    3 CONTINUE
C
C     COMPUTE THE ZEROS OF T(NOSCHB+1,Q)
C
      NDEGRE=NOSCHB+1
      CALL ZEROS(NDEGRE,NZERO,ZERO)
      DO 4 I=1,NDEGRE
      QNOISE(I)=ZERO(I)
    4 CONTINUE
C
C     COMPUTE THE ZEROS OF T(IVELCHB+1,Q)
C
      NDEGRE=IVELCHB+1
      CALL ZEROS(NDEGRE,NZERO,ZERO)
      DO 12 I=1,NDEGRE
      QVEL(I)=ZERO(I)
   12 CONTINUE
C
      IF(IMEAN .NE. 1) GO TO 5
C
C     CALL SUBROUTINE ABSIAS TO OBTAIN ETA(I)
C
      CALL ABSIAS
```

```
C
C       CALL SUBROUTINE PKOAT TO OBTAIN PKO(I,J),
C       PKA(I,J), AND PKT(I,J), I=1(1)25, J=1(1)NNOISE+1
C       IF NTAPE=0
C
        IF(NTAPE.NE.0) GO TO 8
        CALL PKOAT
        GO TO 5
C
C       READ IN PKO(I,J), PKA(I,J), AND PKT(I,J)
C       FROM DATA FILE PKOATD
C
  8     REWIND2
        JEND=NNOISE+1
        DO 9 I=1,25
        DO 10 J=1,JEND
        READ(2,11) PKO(I,J),PKA(I,J),PKT(I,J)
 10     CONTINUE
  9     CONTINUE
 11     FORMAT(3(2X,E23.16))
C
C       CALL SUBROUTINE AERODYN TO OBTAIN THE
C       INITIAL SET OF POINTS USED IN THE
C       SUBROUTINE COMPLEX
C
C
C
  5     IF(IPRBLM.NE.3)
       $ CALL AERODYN(TRYRAM,NRAM,ORAMDA,NOR,NTRY,NTP,CTP)
        IPR=(IPRBLM-1)*(IPRBLM-2)
        IF(IPR.NE.0) CALL NOPTIML(ORAMDA,NOR,NTP,CTP)
        IF(IPRBLM .NE. 4) STOP
C
C       CALL COMPLEX TO OBTAIN THE AEROACOUSTIC
C       OPTIMAL CIRCULATION DISTRIBUTION
C
```

```
      CALL COMPLEX(ORAMDA,NOR)
      CALL RAMCOE(ORAMDA,NOR)
      CALL HATIJ
      CALL CIRCUL
      CALL UATCHBY
40    FORMAT(//,5X,"**********************************")
41    FORMAT(5X,"*****                        *****")
42    FORMAT(5X,"***** AEROACOUSTIC SOLUTION *****")
43    FORMAT(5X,"**********************************"//)
      WRITE(6,40)
      WRITE(6,41)
      WRITE(6,42)
      WRITE(6,41)
      WRITE(6,43)
      WRITE(6,31) MCIRCU
31    FORMAT(5X,"NON-DIMENSIONAL CIRCULATION"//,5X,
     $ "GAMMA(Q)=SQRT(1-Q**2)*SUMMATION G(I)*U(I-1,Q), I=1,"
     $ ,I2//,5X,"WHERE G(I):"//)
      WRITE(6,32) (I,G(I),I=1,MCIRCU)
32    FORMAT(5X,"G(",I2,")=",F25.10)
      CALL AEROCF
      WRITE(6,33) CT,CP,ETAI
33    FORMAT(//,5X,"CT=",F10.5//,5X,"CP=",F10.5//,5X,
     $ "IDEAL EFFICIENCY=",F10.5//)
      WRITE(6,44)
44    FORMAT(///,10X,"++R++",12X,"++Q++",11X,"++UA++",
     $ 11X,"++UT++",9X,"++RAMDAI++",8X,"++GAMMA++"//)
      R=HUB
35    IF(R.GT.1.) GO TO 37
      Q=(2.*R-(1.+HUB))/(1.-HUB)
      CALL VELOCIT(UA,UT,Q)
      RAM=R*(VF+UA)/(R/RAMDAP+UT)
      WRITE(6,36) R,Q,UA,UT,RAM,CIRLF(Q)
      R=R+0.1
      GO TO 35
36    FORMAT(6(2X,F15.7))
```

```
   37 STOP
      END
C
C
C     ************************************
C     *****                          *****
C     *****  COMPLEX    METHOD       *****
C     *****                          *****
C     ************************************
C
C
      SUBROUTINE COMPLEX(ORAMDA,NOR)
C
C
C     PURPOSE
C
C     TO FIND THE AEROACOUSTIC OPTIMAL
C     CIRCULATION
C
C     DESCRIPTION OF PARAMETERS
C
C     SEE SUBROUTINE JCONSX AND MAIN PROGRAM
C
C
      DIMENSION ORAMDA(NOR)
      DIMENSION X(15,15),R(15,15),F(15)
      DIMENSION G(15),H(15),XC(15)
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA16/OZERO(15)
      COMMON /JTRAIN/JYES
      COMMON /DATA1/JRAMDA
      COMMON /DATA6/HUB
      COMMON /DATA15/CT,CP,ETAI
      COMMON /DATA30/ENSMSQ(15)
      COMMON /DATA34/CTRAIN,CPRAIN
      COMMON /DATA37/ENS(2)
```

```
      COMMON /DATA38/XX(2,15)
      COMMON /DATA39/GT(2),GP(2)
      INTEGER GAMMA
C
C***  INPUT PARAMETERS
C
      KO=6
      KX=15
      LX=15
      MX=15
      NX=15
      N=JRAMDA+1
      M=N+1
      L=M
      K=N+2
      ALPHA=1.3
      BETA=1.E-8
      GAMMA=4
      DELTA=0.0001
      ITMAX=140
C
C***  END OF INPUT
C
C
C
   21 FORMAT(//,5X,"**********************************")
   22 FORMAT(5X,"*****                        *****")
   23 FORMAT(5X,"***** BEGIN COMPLEX METHOD *****")
   24 FORMAT(5X,"**********************************"//)
      WRITE(6,21)
      WRITE(6,22)
      WRITE(6,23)
      WRITE(6,22)
      WRITE(6,24)
C
C     INPUT THE INITIAL SET OF POINTS
```

```
C
      DO 1 I=1,N
      X(1,I)=XX(1,I)
      X(2,I)=XX(2,I)
    1 CONTINUE
      X(1,N+1)=GT(1)
      X(1,N+2)=GP(1)
      X(2,N+1)=GT(2)
      X(2,N+2)=GP(2)
C
C     CONSTRAINT ON CT OR CP
C
      IF(GT(1).LE.GT(2))CTRAIN=GT(1)
      IF(GT(1).GE.GT(2))CTRAIN=GT(2)
      IF(GP(1).LE.GP(2))CPRAIN=GP(2)
      IF(GP(1).GE.GP(2))CPRAIN=GP(1)
C
C     CHECK THE INITIAL SET OF POINTS
C
      DO 40 IW=1,2
      IU=IW
      JYES=1
      CALL JCNST1(N,M,K,X,G,H,IU,L,KX,LX,MX,NX)
      DO 2 I=1,M
      IF((X(IW,I)-G(I)).LT.0.) GO TO 3
      IF((H(I)-X(IW,I)).LT.0.) GO TO 3
    2 CONTINUE
   40 CONTINUE
      GO TO 4
    3 WRITE(6,5)
    5 FORMAT(//,5X,"*** EXECUTION IS TERMINATED**"//)
      WRITE(6,6)
    6 FORMAT(5X,"THE INITIAL SET OF POINTS THAT"/,5X,
     $ "DOESE NOT SATISFIES CONSTRAINTS"//)
      DO 8 I=1,M
      WRITE(6,9)IW,I,X(1,I)
```

```
   8  CONTINUE
   9  FORMAT(5X,"X(",I2,",",I2,")=",F15.7)
      STOP
C
C     GIVE ENSMSQ(1) AND ENSMSQ(2)
C
   4  ENSMSQ(1)=ENS(1)
      ENSMSQ(2)=ENS(2)
C
C     INPUT THE RANDOM NUMBERS R(K,N)
C
C     UES SUBPROGRAM FUNCTION RANF(A) TO GENERATE
C     RANDOM NUMBERS
C
      DO 30 I=1,K
      DO 31 J=1,N
      R(I,J)=RANF(AA)
  31  CONTINUE
  30  CONTINUE
C
C     CALL SUBROUTINE JCONSX TO OBTAIN THE
C     AEROACOUSTIC SOLUTION
C
      CALL JCONSX(N,M,K,ITMAX,ALPHA,BETA,GAMMA,
     $ DELTA,X,R,F,IT,IEV2,KO,G,H,XC,L,KX,LX,MX,NX)
      DO 10 I=1,N
      ORAMDA(I)=X(IEV2,I)
  10  CONTINUE
      CT=X(IEV2,N+1)
      CP=X(IEV2,N+2)
      SQMEAN=-F(IEV2)
      ETAI=VF*CT/CP
      WRITE(6,11)
  11  FORMAT(//,5X,"*******************************")
  12  FORMAT(5X,"*****                          *****")
  13  FORMAT(5X,"***** OUTPUT FROM COMPLEX *****")
```

```
18   FORMAT(5X,"********************************")
     WRITE(6,12)
     WRITE(6,13)
     WRITE(6,12)
     WRITE(6,18)
     IDG=JRAMDA+1
     WRITE(6,14)
14   FORMAT(//,5X,"HYDRODYNAMIC ADVANCE COEFFICIENTS"//)
     WRITE(6,20)
20   FORMAT(//,15X,"++R++",12X,"++Q++",9X,"++RAMDAI++"//)
     DO 15 I=1,IDG
     ROU=0.5*((1.-HUB)*OZERO(I)+1.+HUB)
     WRITE(6,16) ROU,OZERO(I),ORAMDA(I)
15   CONTINUE
16   FORMAT(5X,3(2X,F15.7))
     WRITE(6,17) CT,CP,ETAI,SQMEAN
17   FORMAT(//,5X,"CT=",F15.7//,5X,"CP=",F15.7//,5X,
    $ "IDEAL EFFICIENCY=",F15.7//,5X,"ENSEMBLE MEAN SQUARE=",
    $ E25.12//)
     RETURN
     END
C
C
     SUBROUTINE MEANSQF(X,I,N,KX,LX)
C
C    PURPOSE
C
C    TO COMPUTE THE OBJECTIVE FUNCTION
C    BY CALLING SUBROUTINE MEANSQ AND STORE
C    THE RESULT IN ARRAY ENSMSQ(I) FOR THE
C    ITH SET OF POINTS
C
C
     DIMENSION X(KX,LX)
     COMMON /DATA30/ENSMSQ(15)
     COMMON /DATA24/SQMEAN
```

```
      CALL MEANSQ
      ENSMSQ(I)=SQMEAN
      RETURN
      END
C
C
      SUBROUTINE JFUNC(N,M,K,X,F,I,L,KX,LX,MX,NX)
C
C
C     PURPOSE
C
C
C     TO SUPPLY THE FUNCTIONAL VALUE
C     OF THE OBJECTIVE FUNCTION THAT IS
C     TO BE MAXIMIZED
C
C
      DIMENSION X(KX,LX),F(KX)
      COMMON /DATA30/ENSMSQ(15)
      F(I)=-ENSMSQ(I)
      RETURN
      END
C
C
      SUBROUTINE AECOEF(N,M,K,X,I,KX,LX)
C
C     PURPOSE
C
C     TO COMPUTE CT AND CP FOR A GIVEN SET OF POINTS
C
      DIMENSION ORAMDA(15),X(KX,LX)
      COMMON/DATA1/JRAMDA
      NOR=15
      JX=JRAMDA+1
      DO 1 J=1,JX
      ORAMDA(J)=X(I,J)
```

```
    1  CONTINUE
       CALL RAMCOE(ORAMDA,NOR)
       CALL HATIJ
       CALL CIRCUL
       CALL UATCHBY
       CALL AEROCF
       RETURN
       END
C
C
       SUBROUTINE JCNST1(N,M,K,X,G,H,I,L,KX,
      $ LX,MX,NX)
C
C
C      PURPOSE
C
C
C      TO PROVIDE THE CONSTRAINTS
C
C
       DIMENSION X(KX,LX),G(MX),H(MX)
       COMMON /DATA15/CT,CP,ETAI
       COMMON /JTRAIN/JYES
       COMMON /DATA34/CTRAIN,CPRAIN
C
C
C***   INPUT THE CONSTRAINTS
C
C      EXPLICIT CONSTRAINTS (CONSTRAINTS ON RAMDAI)
C
       DO 1 ITAB=1,N
       G(ITAB)=0.1
       H(ITAB)=0.6
    1  CONTINUE
C
C***   END OF INPUT OF EXPLICIT CONSTRAINTS
```

```
C
      IF(JYES .NE. 1) RETURN
C
C     TO PROVIDE THE VALUES AND CONSTRAINTS
C     OF THE IMPLICIT VARIABLES
C
C
C     CALL AECOEF TO OBTAIN CT AND CF
C
      CALL AECOEF(N,M,K,X,I,KX,LX)
      X(I,N+1)=CT
      X(I,N+2)=CP
C
C***  INPUT THE IMPLICIT CONSTRAINTS
C     (CONSTRAINTS ON CT AND CP)
C
      G(N+1)=CTRAIN-1.E-6
      H(N+1)=CT+1.
      G(N+2)=0.5
      H(N+2)=5.
C
C***  END OF INPUT OF IMPLICIT CONSTRAINTS
C
C
C***  END OF INPUT OF CONSTRAINTS
C
      RETURN
      END
C
C
C
C
      SUBROUTINE JCONSX(N,M,K,ITMAX,ALPHA,BETA,GAMMA,
     $ DELTA,X,R,F,IT,IEV2,KO,G,H,XC,L,KX,LX,MX,NX)
C
C
```

```
C      THIS IS A MODIFIED PROGRAM OF RICHARDSON AND
C      KUESTERS' ORIGINAL PROGRAM (COMM. ACM 16,
C      AUG, L973, PP. 487-489)
C      THIS MODIFIED PROGRAM USES TWO INITIAL
C      FEASIBLE SOLUTIONS (AERODYNAMIC OPTIMUM SOLUTION
C      AND NON-OPTIMUM SOLUTION) AS INPUT AND
C      COMPUTE THE IMPLICIT VARIABLES ONLY WHEN THEY ARE
C      NEEDED
C
C      PURPOSE
C
C      TO FIND THE CONSTRAINED MAXIMUM OF A FUNCTION OF
C      SEVERAL VARIABLES BY THE COMPLEX METHOD OF M. J. BOX.
C      THIS PROGRAM IS WRITTEN BY JOEL A. RICHARDSON AND
C      J. L. KUESTER, COMM. ACM16(AUG,1973),(487-489)
C      THIS IS THE PRIMARY SUBROUTINE AND COORDINATES THE
C      SPECIAL PURPOSE SUBROUTINES (JCEK1, JCENT,
C      JFUNC, JCNST1).. INITIAL GUESSES OF THE INDEPENDENT
C      VARIABLES, RANDAOM NUMBERS,
C      SOLUTION PARAMETERS, DIMENSION LIMITS AND PRINTER CODE
C      DESTINATION ARE OBTAINED FROM THE MAIN PROGRAM
C      (SUBROUTINE COMPLEX).
C      FINAL FUNCTION AND INDEPENDENT VARIABLE VALUES ARE
C      TRANSFERRED TO THE MAIN PROGRAM FOR PRINTOUT.
C      INTERMINATE PRINTOURS ARE PROVIDED IN THIS SUBROUTINE.
C      THE USER MUST PROVIDE THE MAIN PROGRAM AND THE
C      SUBROUTINES THAT SPECIFY THE FUNCTION (JFUNC)
C      AND CONSTRAINTS (JCNST1).
C      FORMAT CHANGES MAY BE REQUIRED WITHIN THIS SUBROUTINE
C      DEPENDING ON THE PARTICULAR PROBLEM UNDER CONSIDERATION.
C      USAGE
C      CALL JCONSX(N,M,K,ITAX,ALPHA,BETA,GAMMA,DELTA,X,R,F,IT
C      ,IEV2,KO,G,H,XC,L,KX,LX,MX,NX)
C      SUBROUTINE REQUIRED
C      JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,KX,LX,MX,NX)
C        CHECKS ALL POINTS AGAINST EXPLICIT AND IMPLICIT
```

```
C      CONSTRAINTS AND APPLIES CORRECTION IF VIOLATIONS ARE
C      FOUND
C      JCENT(N,M,K,IEV1,I,XC,X,L,K1,KX,LX,MX,NX)
C        CALCULATES THE CENTROID OF POINTS
C      JFUNC(N,M,K,X,F,I,L,KX,LX,MX,NX)
C      SPECIFIES THE OBJECTIVE FUNCTION (USER SUPPLIED)
C      JCNST1(N,M,K,X,G,H,I,L,KX,LX,MX,NX)
C          SPECIFIES EXPLICIT AND IMPLICIT CONSTRAINT LIMITS
C      (USER SUPPLIED). ORDER EXPLICIT CONSTRAINTS FIRST
C      DESCRIPTION OF THE PARAMETERS
C        KX    NUMBER OF ROWS IN THE DIMENSION STATEMENT
C              FOR X, F, AND R,KX.GE.K-DEFINE IN MAIN PROGRAM
C              (USER SUPPLIED)
C        LX    NUMBER OF COLUMNS IN THE DIMENSION STATEMENT
C              FOR X, LX.GE.L-DEFINE IN MAIN PROGRAM (USER
C              SUPPLIED)
C        MX    DIMENSION OF G AND H,MX.GE.M-DEFINE IN MAIN
C              PROGRMA (USER SUPPLIED)
C        NX    DIMENSION OF XC AND THE NUMBER OF COLUMNS
C              NX.GE.N-DEFINE IN MAIN PROGRAM (USER SUPPLIED)
C        N     NUMBER OF EXPLICIT INDEPENDENT VARIABLES-
C              DEFINED IN THE MAIN PROGRAM
C        M     NUMBER OF SETS OF CONSTRAINTS -DEFINED IN
C              THE MAIN PROGRAM
C        K     NUMBER OF POINTS IN THE COMPLEX -DEFINED IN
C              THE MAIN PROGRAM
C      ITMAX   MAXIMUM NUMBER OF ITERATIONS -DEFINED IN MAIN
C              PROGRAM
C      ALPHA   REFLECTION FACTOR - DEFINED IN MAIN PROGRAM
C      BETA    CONVERGENCE PARAMETER -DEFINED IN MAIN PROGRAM
C      GAMMA   CONVERGENCE PARAMETER - DEFINED IN MAIN PROGRAM
C      DELTA   EXPLICIT CONSTRAINT VIOLATION
C              -DEFINED IN MAIN PROGRAM
C        X     INDEPENDENT VARIABLES - DEFINED IN MAIN PROGRAM
C        R     RANDOM NUMBERS BETWEEN 0 AND 1 - DEFINED IN MAIN
C              PROGRAM
```

```
C          F     OBJECTIVE FUNCTION -DEFINED IN SUBROUTINE JFUNC
C          IT    ITERATION INDEX -DEFINED IN SUBROUTINE  JCONSX
C        IEV2    INDEX OF POINT WITH MAXIMUN FUNCTION VALUV
C                 -DEFINED IN SUBROUTINE JCONSX
C        IEV1    INDEX OF POINT WITH MINIMUM FUNCTION VALUE
C                -DEFINED IN SUBROUTINE JCONSX
C         KO     PRINTER UNIT NUMBER -DEFINED IN MAIN PROGRAM
C          G     LOWER CONSTRAINT -DEFINED IN SUBROUTINE JCNST1
C          H     UPPER CONSTRAINT -DEFINED IN SUBROUTINE JCNST1
C         XC     CENTROID -DEFINED IN SUBROUTINE JCENT
C          L     TOTAL NUMBER OF INDEPENDENT VARIABLES (EXPLICIT +
C                IMPLICIT) -DEFINED IN MAIN PROGRAM
C          I     POINT INDEX -DEFINED IN SUBROUTINE JCONSX
C         KODE   KEY USED TO DETERMINE IF IMPLICIT CONSTRAINTS
C                ARE PROVIDED -DEFINED IN SUBROUTINE
C                JCONSX AND JCEK1
C          K1    DO LOOP LIMIT -DEFINED IN SUBROUTINE JCONSX
C         JYES   IF JYES=1, IMPLICIT VARIABLES ARE COMPUTED
C                IN SUBROUTINE JCNST1
C                IF JYES .NE. 1 IMPLICIT VARIABLES ARE NOT COMPUTED
C
C
C
      DIMENSION X(KX,LX),R(KX,NX),F(KX),G(MX),H(MX),XC(NX)
      INTEGER GAMMA
      COMMON /JTRAIN/JYES
      JYES=2
      IT=1
      WRITE(KO,99995) IT
      KODE=0
      IF (M-N) 20,20, 10
   10 KODE=1
   20 CONTINUE
      DO 40 II=3,K
      DO 30 J=1,N
      X(II,J)=0.
```

```
   30   CONTINUE
   40   CONTINUE
C       CALCULATE COMPLEX POINTS AND CHECK AGAINST CONSTRAINTS
        DO 60 II=3,K
         DO 50 J=1,N
            I=II
            CALL JCNST1(N,M,K,X,G,H,I,L,KX,LX,MX,NX)
            X(II,J)=G(J)+R(II,J)*(H(J)-G(J))
   50   CONTINUE
        K1=II
        CALL JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,
       $ KX,LX,MX,NX)
        WRITE(KO,99999)II,(X(II,J),J=1,N)
   60   CONTINUE
        K1=K
        DO 70 I=1,K
        CALL JFUNC(N,M,K,X,F,I,L,KX,LX,MX,NX)
   70   CONTINUE
        KOUNT=1
        IA=0
C       FIND POINT WITH LOWEST FUNCTION VALUE
        WRITE(KO,99998)  (F(I),I=1,K)
        WRITE(KO,99993) (X(I,N+1),I=1,K)
        WRITE(KO,99994) (X(I,N+2),I=1,K)
   80   IEV1=1
        DO 100 ICM=2,K
        IF(F(IEV1)-F(ICM))100,100,90
   90   IEV1=ICM
  100   CONTINUE
C       FIND POINT WITH HIGHTEST FUNCTION VALUE
        IEV2=1
        DO 120 ICM=2,K
        IF(F(IEV2)-F(ICM)) 110,110,120
  110   IEV2=ICM
  120   CONTINUE
C       CHECK CONVERGENCE CRITERIA
```

```
      IF(F(IEV2)-(F(IEV1)+BETA)) 140,130,130
  130 KOUNT=1
      GO TO 150
  140 KOUNT=KOUNT+1
      IF(KOUNT-GAMMA)150,240,240
C     REPLACE POINT BY LOWEST FUNCTION VALUE
  150 CALL JCENT(N,M,K,IEV1,I,XC,X,L,K1,KX,LX,MX,NX)
      DO 160 J=1,N
      X(IEV1,J)=(1.+ALPHA)*(XC(J))-ALPHA*(X(IEV1,J))
  160 CONTINUE
      I=IEV1
      CALL JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,
     $ KX,LX,MX,NX)
      CALL JFUNC(N,M,K,X,F,I,L,KX,LX,MX,NX)
C     REPLACE NEW POINT IF IT REPEATS AS LOWEST FUNCTION VALUE
  170 IEV2=1
      DO 190 ICM=2,K
      IF(F(IEV2)-F(ICM))190,190,180
  180 IEV2=ICM
  190 CONTINUE
      IF(IEV2-IEV1)220,200,220
  200 DO 210 JJ=1,N
      X(IEV1,JJ)=(X(IEV1,JJ)+XC(JJ))/2.
  210 CONTINUE
      I=IEV1
      CALL JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,
     $ KX,LX,MX,NX)
      CALL JFUNC(N,M,K,X,F,I,L,KX,LX,MX,NX)
      GO TO 170
  220 CONTINUE
      WRITE(KO,99997)  (X(IEV1,JB),JB=1,N)
      WRITE(KO,99998)  (F(I),I=1,K)
      WRITE(KO,99993) (X(I,N+1),I=1,K)
      WRITE(KO,99994) (X(I,N+2),I=1,K)
      WRITE(KO,99996) (XC(J),J=1,N)
      IT=IT+1
```

```
      IF(IT-ITMAX) 230,230,240
 230  CONTINUE
      WRITE(KO,99995) IT
      GO TO 80
 240  RETURN
99999  FORMAT(1H , 15X, 21H COORDINATES AT POINT,
     $ I4/5(F15.7,2X))
99998  FORMAT(1H , 20X, 16H FUNCTION VALUES, /5(E15.7,2X))
99997  FORMAT(1H , 20X,16H CORRECTED POINT, /5(F15.7,2X))
99996  FORMAT(1H , 21H CENTROID COORDINATES, 2X, 5(F15.7,2X))
99995  FORMAT(1H , //10H ITERATION,4X,I5)
99994  FORMAT(1H , 20X, "POWER COEFFICIENTS",/5(F15.7,2X))
99993  FORMAT(1H , 20X, "THRUST COEFFICIENTS",/5(F15.7,2X))
      END
C
C
      SUBROUTINE JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,
C
C
     $ KX,LX,MX,NX)
C      PURPOSE
C      TO CHECK ALL POINTS AGAINST THE EXPLICIT AND IMPLICIT
C      CONSTRAINTS AND TO APPLY CORRECTIONS IF VIOLATIONS
C      ARE FOUND
C
C
C      USAGE
C
C
C      CALL JCEK1(N,M,K,X,G,H,I,KODE,XC,DELTA,L,K1,KX,LX,MX,NX)
C      SUBROUTINE REQUIRED
C      JCENT(N,M,K,IEV1,I,XC,L,K1,KX,LX,MX,NX)
C      JCNST1(N,M,K,X,G,H,I,L,KX,LX,MX,NX)
C      DESCRIPTION OF PARAMETERS
C      PREVIOUSLY DEFINED IN SUBROUTINE JCONSX
      DIMENSION X(KX,LX),G(MX),H(MX),XC(NX)
```

```
      COMMON /JTRAIN/JYES
   10 KT=0
      JYES=2
      CALL JCNST1(N,M,K,X,G,H,I,L,KX,LX,MX,NX)
C     CHECK AGAINST EXPLICIT CONSTRAINTS
      DO 50 J=1,N
      IF(X(I,J)-G(J)) 20 ,20, 30
   20 X(I,J)=G(J)+DELTA
      GO TO 50
   30 IF(H(J)-X(I,J)) 40,40,50
   40  X(I,J)=H(J)-DELTA
   50 CONTINUE
      IF(KODE)110,110,60
C     CHECK AGAINSTTHE IMPLICIT CONSTRAINTS
   60 CONTINUE
      NN=N+1
      JYES=1
      DO 100 J=NN,M
      CALL JCNST1(N,M,K,X,G,H,I,L,KX,LX,MX,NX)
      IF(X(I,J)-G(J)) 80,70,70
   70 IF(H(J)-X(I,J)) 80,120,120
   80 IEV1=I
      KT=1
      CALL JCENT(N,M,K,IEV1,I,XC,X,L,K1,KX,LX,MX,NX)
      DO 90 JJ=1,N
      X(I,JJ)=(X(I,JJ)+XC(JJ))/2.
   90 CONTINUE
      JYES=1
      GO TO 100
  120 JYES=2
  100 CONTINUE
      IF(KT) 110,110,10
  110 CALL MEANSQF(X,I,N,KX,LX)
      JYES=2
      RETURN
      END
```

```
C
C
      SUBROUTINE JCENT(N,M,K,IEV1,I,XC,X,L,K1,KX,LX,MX,NX)
C
C
C     PURPOSE
C     TO CALCULATE THE CENTROID OF POINTS
C     USAGE
C     CALL JCENT(N,M,K,IEV1,I,XC,X,L,K1,KX,LX,MX,NX)
C     SUBROUTINE REQUIRED
C     NONE
      DIMENSION X(KX,LX),XC(NX)
      DO 20 J=1,N
      XC(J)=0.
      DO 10 IL=1,K1
      XC(J)=XC(J)+X(IL,J)
   10 CONTINUE
      RK=K1
      XC(J)=(XC(J)-X(IEV1,J))/(RK-1.)
   20 CONTINUE
      RETURN
      END
C
C
C     *****************************
C     *****                   *****
C     ***** AERODYNAMIC MODEL *****
C     *****                   *****
C     *****************************
C
C
      SUBROUTINE AERODYN(TRYRAM,NRAM,ORAMDA,
     $ NOR,NTRY,NTP,CTP)
C
C
C     PURPOSE
```

```
C
C     TO ESTIMATE THE FIRST APPROXIMATION OF THE
C     HYDRODYNAMIC ADVANE COEFFICIENT BY
C     SOLVING AERODYNAMIC OPTIMAL PROBLEM
C
C
C     DESCRIPTION OF PARAMETERS
C
C       TRYRAM(I)       THE GUESSED VALUES OF RAMDAI
C       NTP              NTP=1 IF CT IS SPECIFIED
C
C                        NTP .NE. 1 IF CP IS SPECIFIED
C      CTP              CTP=CT IF NTP=1
C                       CTP=CP IF NTP .NE.1
C       NTRY            NUMBER OF ELEMENTS GIVEN IN TRYRAM(I)
C       NOR             LENGTH OF ORAMDA(I)
C       ORAMDA(I)       HYDRODYNAMIC ADVANCE COEFFICIENT AT
C                       THE ZEROS OF T(JRAMDA+1,Q)
C       NRAM            LENGTH OF TRYRAM(I)
C
C
      DIMENSION TRYRAM(NRAM),ORAMDA(NOR),X(5)
      COMMON /DATA1/JRAMDA
      COMMON /DATA6/HUB
      COMMON /DATA11/MCIRCU
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA14/G(15)
      COMMON /DATA15/CT,CP,ETAI
      COMMON /DATA24/SQMEAN
      COMMON /DATA36/IMEAN,IPRBLM
      COMMON /DATA37/ENS(2)
      COMMON /DATA38/XX(2,15)
      COMMON /DATA39/GT(2),GP(2)
      JX=JRAMDA+1
      IXX=NTRY+1
 40   FORMAT(//,5X,"***********************************")
```

```
 41   FORMAT(5X,"*****                                  *****")
 42   FORMAT(5X,"***** OUTPUT FROM AERODYNAMIC   *****")
 43   FORMAT(5X,"***** OPTIMUM PROBLEM                *****")
 44   FORMAT(5X,"*************************************")
      WRITE(6,40)
      WRITE(6,41)
      WRITE(6,42)
      WRITE(6,43)
      WRITE(6,41)
      WRITE(6,44)
      DO 51 II=1,IXX
C
C     COMPUTE THE FIRST GUESS OF RAMDA(OQZERO(I))
C     THAT IS, ORAMDA(I)
C
      DO 52 J=1,JX
      ORAMDA(J)=TRYRAM(II)
 52   CONTINUE
      WRITE(6,56) ORAMDA(1)
 56   FORMAT(//,5X,"RAMDAI=",F15.10//)
      CALL RAMCOE(ORAMDA,NOR)
      CALL HATIJ
      CALL CIRCUL
      CALL UATCHBY
      WRITE(6,5) (I,G(I),I=1,MCIRCU)
  5   FORMAT(5X,"G(",I2,")=",F25.10)
      CALL AEROCF
      WRITE(6,6)CT,CP,ETAI
  6   FORMAT(//,5X,"CT=",F10.5//5X,"CP=",F10.5//,5X,
     $ "IDEAL EFFICIENCY ETAI=",F10.5//)
      WRITE(6,57)
 57   FORMAT(//,10X,"++R++",12X,"++Q++",11X,"++UA++",
     $ 11X,"++UT++",9X,"++RAMDAI++",8X,"++GAMMA++"//)
      R=HUB
  7   IF(R.GT.1.) GO TO 9
      Q=(2.*R-(1.+HUB))/(1.-HUB)
```

```
      CALL VELOCIT(UA,UT,Q)
      RAM=R*(VF+UA)/(R/RAMDAP+UT)
      WRITE(6,8) R,Q,UA,UT,RAM,CIRLF(Q)
      R=R+0.1
      GO TO 7
    8 FORMAT(6(2X,F15.7))
    9 IF(NTRY.EQ.1) GO TO 55
      IF(II.EQ.IXX) GO TO 55
      IF(NTP.EQ.1) X(II)=CT
      IF(NTP.NE.1) X(II)=CP
      IF(II.LT.NTRY) GO TO 51
C
C     INTERPOLATE THE OPTIMUM HYDRODYNAMIC
C     ADVANCE COEFFICIENT
      Y=CTP
      SUM=0.
      DO 53 I=1,NTRY
      A=1.
      B=1.
      DO 54 J=1,NTRY
      IF(J.EQ.I) GO TO 54
      A=A*(Y-X(J))
      B=B*(X(I)-X(J))
   54 CONTINUE
      SUM=SUM+(A/B)*TRYRAM(I)
   53 CONTINUE
      TRYRAM(NTRY+1)=SUM
   51 CONTINUE
   55 IF(IMEAN.NE.1) RETURN
      CALL MEANSQ
      ENS(1)=SQMEAN
      WRITE(6,60) ENS(1)
   60 FORMAT(//5X,"ENSEMBLE MEAN SQUARE OF"/,5X,
     $ "THE ACOUSTIC PRESSURE:"/,5X,E25.12//)
      IF(IPRBLM .NE. 4) RETURN
      GT(1)=CT
```

```
      GP(1)=CP
      DO 61 J=1,JX
      XX(1,J)=ORAMDA(J)
   61 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE NOPTIML(ORAMDA,NOR,NTP,CTP)
C
C     PURPOSE
C
C     TO OBTAIN THE SECOND FEASIBLE SOLUTION OF
C     HYDRODYNAMIC ADVANCE COEFFICIENT BY SOLVING THE
C     NON-OPTIMUM PROBLEM WITH GIVEN CHARACTERISING
C     FUNCTION
C
C
      REAL KTP
      DIMENSION ORAMDA(NOR),GG(15),RAM(15)
      COMMON /DATA1/JRAMDA
      COMMON /DATA6/HUB
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /CONST/PAI
      COMMON /DATA11/MCIRCU
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA14/G(15)
      COMMON /DATA15/CT,CP,ETAI
      COMMON /DATA16/OQZERO(15)
      COMMON /DATA35/CT1,CT2,CP1,CP2
      COMMON /DATA24/SQMEAN
      COMMON /DATA36/IMEAN,IPRBLM
      COMMON /DATA37/ENS(2)
      COMMON /DATA38/XX(2,15)
      COMMON /DATA39/GT(2),GP(2)
   20 FORMAT(//,5X,"****************************")
```

```
   21  FORMAT(5X,"*****                           *****")
   22  FORMAT(5X,"***** OUTPUT FROM NON- *****")
   23  FORMAT(5X,"***** OPTIMUM PROBLEM   *****")
   24  FORMAT(5X,"*****************************"//
       WRITE(6,20)
       WRITE(6,21)
       WRITE(6,22)
       WRITE(6,23)
       WRITE(6,21)
       WRITE(6,24)
C
C***   INPUT DATA
C
C
C      INPUT THE MAXIMUN NUMBER OF ITERATIONS
C
       IMAX=30
C
C      INPUT THE COEFFICIENTS OF CHARACTERISING FUNCTION
C
       GG(1)=0.03267
       GG(2)=-0.00083
       GG(3)=-0.00467
       GG(4)=-0.00243
       GG(5)=-0.00027
       IF(MCIRCU.LE.5) GO TO 1
       DO 2 M=6,MCIRCU
       GG(M)=0.
    2  CONTINUE
    1  DO 3 M=1,MCIRCU
       G(M)=GG(M)
    3  CONTINUE
C
C
C***   END OF INPUT
C
```

```
C
C     FIRST APPROXIMATION
C
      JMAX=JRAMDA+1
      DO 4 J=1,JMAX
      ORAMDA(J)=RAMDAP*VF
    4 CONTINUE
      I=1
      WRITE(6,5)I
    5 FORMAT(/5X,"ITERATION",2X,I3/)
    6 CALL RAMCOE(ORAMDA,NOR)
      CALL HATIJ
      CALL UATCHBY
      CALL CTP12
      IF(NTP.NE.1) GO TO 7
      KTP=(-CT1+SQRT(CT1*CT1+4.*CTP*CT2))/
     $ (2.*CT2)
      GO TO 8
    7 KTP=(-CP1+SQRT(CP1*CP1+4.*CTP*CP2))/
     $ (2.*CP2)
C
C     COMPUTE NEW RAMDAI AT OQZERO(I)
C
      ICHK=2
    8 DO 9 J=1,JMAX
      Q=OQZERO(J)
      R=0.5*((1.-HUB)*Q+1.+HUB)
      CALL VELOCIT(UA,UT,Q)
      RAM(J)=R*(VF+KTP*UA)/(R/RAMDAP+
     $ KTP*UT)
    9 CONTINUE
      DO 11 J=1,JMAX
      IF(ABS(RAM(J)-ORAMDA(J)).GT.1.E-6) GO TO 12
   11 CONTINUE
      DO 13 J=1,JMAX
      ORAMDA(J)=0.5*(ORAMDA(J)+RAM(J))
```

```
13   CONTINUE
     GO TO 10
12   DO 14 J=1,JMAX
     ORAMDA(J)=C.5*(ORAMDA(J)+RAM(J))
14   CONTINUE
     I=I+1
     IF(I.GT.IMAX) GO TO 10
     WRITE(6,5)I
     WRITE(6,15) KTP
15   FORMAT(5X,"KTP=",F15.8/)
     WRITE(6,16) (ORAMDA(J),J=1,JMAX)
16   FORMAT(5X,"VALUES OF RAMDAI",/5(F15,8,2X))
     GO TO 6
10   DO 17 M=1,MCIRCU
     G(M)=KTP*GG(M)
17   CONTINUE
     CALL UATCHBY
     CT=KTP*CT1+KTP*KTP*CT2
     CP=KTP*CP1+KTP*KTP*CP2
     ETAI=CT*VF/CP
     WRITE(6,35) (I,G(I),I=1,MCIRCU)
35   FORMAT(5X,"G(",I2,")=",F25.10)
     WRITE(6,36)CT,CP,ETAI
36   FORMAT(//,5X,"CT=",F10.5//5X,"CP=",F10.5//,5X,
    $ "IDEAL EFFICIENCY ETAI=",F10.5//)
     WRITE(6,57)
57   FORMAT(//,10X,"++R++",12X,"++Q++",11X,"++UA++",
    $ 11X,"++UT++",9X,"++RAMDAI++",8X,"++GAMMA++"//)
     R=HUB
37   IF(R.GT.1.) GO TO 39
     Q=(2.*R-(1.+HUB))/(1.-HUB)
     CALL VELOCIT(UA,UT,Q)
     RAMD=R*(VF+UA)/(R/RAMDAP+UT)
     WRITE(6,38)R,Q,UA,UT,RAMD,CIRLF(Q)
     R=R+0,1
     GO TO 37
```

```
   38  FORMAT(6(2X,F15.7))
   39  IF(IMEAN.NE.1) RETURN
       CALL MEANSQ
       ENS(2)=SQMEAN
       WRITE(6,40) ENS(2)
   40  FORMAT(//,5X,"ENSEMBLE MEAN SQUARE OF"/,5X,
      $ "THE ACOUSTIC PRESSURE:",2X,E25.12//)
       IF(IPRBLM .NE. 4) RETURN
       DO 41 J=1,JMAX
       XX(2,J)=ORAMDA(J)
   41  CONTINUE
       GT(2)=CT
       GP(2)=CP
       RETURN
       END
C
C
       SUBROUTINE CTP12
C
C      PURPOSE
C
C      TO COMPUTE CT1, CT2, CP1, AND CP2
C
       COMMON /CONST/PAI
       COMMON /DATA6/HUB
       COMMON /DATA8/NBLADE,KWENH,LUPIND
       COMMON /DATA11/MCIRCU
       COMMON /DATA12/FM,TM,VF,RAMDAP
       COMMON /DATA14/G(15)
       COMMON /DATA15/CT,CP,ETAI
       COMMON /DATA31/IVELDG,IVELCHB
       COMMON /DATA33/UACHB(51),UTCHB(51)
       COMMON /DATA35/CT1,CT2,CP1,CP2
       NB=NBLADE
C
C      CALCULATE CT1 AND CP1
```

```
C
      FACT=0.5*PAI*(1.-HUB)*FLOAT(NB)
      CT1=FACT*(G(1)*(1.+HUB)+G(2)*0.5 *(1.-HUB))/
     $ RAMDAP
      CP1=FACT*(G(1)*(1.+HUB)*VF+G(2)* 0.5 *(1.-HUB)*
     $ VF)/RAMDAP
C
C     COMPUTE CT2 AND CP2
      CT2=0.
      CP2=0.
      FX=0.5*(1.+HUB)
      FY=0.25*(1.-HUB)
      DO 1 M=1,MCIRCU
      CT2=CT2+G(M)*(UTCHB(M)-UTCHB(M+2))
      CP2=CP2+G(M)*((UACHB(M)-UACHB(M+2))*FX+
     $ (UACHB(IABS(M-2)+1)-UACHB(M+3))*FY)
    1 CONTINUE
      CT2=FACT*CT2
      CP2=FACT*CP2/RAMDAP
      RETURN
      END
D.
C
      SUBROUTINE AEROCF
C
C
C     PURPOSE
C
C     TO COMPUTE THE THRUST AND POWER COEFFICIENTS
C
C
      COMMON /CONST/PAI
      COMMON /DATA6/HUB
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /DATA11/MCIRCU
      COMMON /DATA12/FM,TM,VF,RAMDAP
```

```
      COMMON /DATA14/G(15)
      COMMON /DATA15/CT,CP,ETAI
      COMMON /DATA31/IVELDG,IVELCHB
      COMMON /DATA33/UACHB(51),UTCHB(51)
      NB=NBLADE
      FACT=0.5*PAI*(1.-HUB)*FLOAT(NB)
      FX=0.5*(1.+HUB)
      FY=0.25*(1.-HUB)
      SUMT=G(1)*(1.+HUB)/RAMDAP
      SUMT=SUMT+G(2)*(1.-HUB)/(2.*RAMDAP)
      SUMP=G(1)*(1.+HUB)*VF
      SUMP=SUMP+G(2)*(1.-HUB)*VF/2.
      DO 1 M=1,MCIRCU
      SUMT=SUMT+G(M)*(UTCHB(M)-UTCHB(M+2))
      SUMP=SUMP+G(M)*(FX*(UACHB(M)-UACHB(M+2))+
     $ FY*(UACHB(IABS(M-2)+1)-UACHB(M+3)))
    1 CONTINUE
      CT=SUMT*FACT
      CP=SUMP*FACT/RAMDAP
      ETAI=CT*VF/CP
      RETURN
      END
C
C
      FUNCTION RAIJKM(I,J,K,M)
C
C
C     PURPOSE
C
C     TO COMPUTE RAMAD(I)(J)(K)(M)
C
C
      COMMON /CONST/PAI
      ISUM=0
      MJ=M-J
      KI=K-I
```

```
      IABSMJ=IABS(MJ)
      IABSKI=IABS(KI)
      IF(MJ.NE.0) MJ=MJ/IABSMJ
      IF(KI.NE.0) KI=KI/IABSKI
      IF((I+K-J-M).EQ.0) ISUM=1
      IF((I+K-IABSMJ) .EQ.0) ISUM=ISUM+MJ
      IF((IABSKI-J-M) .EQ.0) ISUM=ISUM+KI
      IF(IABSKI.EQ.IABSMJ) ISUM=ISUM+KI*MJ
      RAIJKM=PAI*FLOAT(ISUM)/8.
      RETURN
      END
C
C
      SUBROUTINE CIRCUL
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS OF THE CHEBYSHEV
C     EXPANSION OF THE CIRCULATION
C
C     CIRCULATION(Q)=SQRT(1-Q**2) *SUMMATION G(M)*
C     U(M-1,Q), M=1(1)MCIRCU
C
C
C     DESCRIPTION OF PARAMETERS
C
C       NT      LENGTH OF TF
C       NU      LENGTH OF UF
C
C
      COMMON /CONST/PAI
      COMMON /DATA3/IDEG,ICHEBY
      COMMON /DATA4/HA(31,31),HT(31,31)
      COMMON /DATA6/HUB
      COMMON /DATA10/QCIRCU(15)
```

```
      COMMON /DATA11/MCIRCU
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA14/G(15)
      DIMENSION A(15,15),B(15,1),RAMDAL(15)
      DIMENSION WK(300),TF(31),UF(31)
C     COMPUTE RAMDA AT QCIRCU(L)
      DO 1 L=1,MCIRCU
      Q=QCIRCU(L)
      RAMDAL(L)=RAMDAF(Q)
    1 CONTINUE
C     CONSTRUCT THE COLUMN VECTOR B
      DO 2 L=1,MCIRCU
      B(L)=RAMDAL(L)/RAMDAP-VF
    2 CONTINUE
C     CONSTRUCT THE COEFFICIENT MATRIX A
      IF(MCIRCU.GE.IDEG) IMAX=MCIRCU
      IF(MCIRCU.LE.IDEG) IMAX=IDEG
      DO 3 L=1,MCIRCU
      Q=QCIRCU(L)
C
C     INPUT DATA
C
      NT=31
      NU=31
C
      CALL TCHEBY(TF,IMAX,Q,NT)
      CALL UCHEBY(UF,IMAX,Q,NT)
      FA=2.*RAMDAL(L)/((1.-HUB)*QCIRCU(L)+1.+HUB)
      DO 5 M=1,MCIRCU
      FAAM=PAI*FLOAT(M)/(1.-HUB)
      SUM=0.
      I=0
    6 HI=1
      IF(I.GT.IDEG) GO TO 7
      IF(I.EQ.0) HI=2.
      SA=0.
```

```
      J=0
    8 HJ=1.
      IF(J.GT.IDEG) GO TO 9
      IF(J.EQ.0) HJ=2.
      IF(J.GT.M) GO TO 10
      SA=SA+(HA(I+1,J+1)-FA*HT(I+1,J+1))*TF(I+1)*
     $ TF(J+1)*UF(M)*FAAM/(HI*HJ)
      J=J+1
      GO TO 8
   10 SA=SA+(HA(I+1,J+1)-FA*HT(I+1,J+1))*TF(I+1)*
     $ TF(M+1)*UF(J)*FAAM/(HI*HJ)
      J=J+1
      GO TO 8
    9 SUM=SUM+SA
      I=I+1
      GO TO 6
    7 A(L,M)=SUM
    5 CONTINUE
    3 CONTINUE
C
C     SOLVE THE SYSTEM OF EQUATIONS A(L,M)*G(M)=B(L)
C     FOR G(M) ,L,M=1(1)MCIRCU
C     BY USING THE LIBRARY SUBROUTINE LEQT2F OF U OF I
C
C
      IER=500
      IDGT=0
      M=1
      N=MCIRCU
      IA=15
      CALL LEQT2F(A,M,N,IA,B,IDGT,WK,IER)
      DO 20 M=1,MCIRCU
      G(M)=B(M,1)
   20 CONTINUE
      IF(((IER-34)*(IER-129)*(IER-131)).NE.0) RETURN
      WRITE(6,24)
```

```
 24  FORMAT(///,5X,"*** ITERATION IS TERMINATED "/,
    $ 5X,"DUE TO ILL-CONDITIONED MATRIX ***")
      STOP
      END
C
C
      SUBROUTINE UATCHBY
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS OF THE CHEBYSHEV
C     EXPANSIONS OF THE INDUCED VELOCITY COMPONENTS
C
C
      COMMON /DATA31/IVELDG,IVELCHB
      COMMON /DATA32/QVEL(51)
      COMMON /DATA33/UACHB(51),UTCHB(51)
      DIMENSION A(50),UA(51),UT(51)
C
C     COMPUTE THE INDUCED VELOCITY COMPONENTS
C
      JMAX=IVELCHB+1
      DO 1 J=1,JMAX
      Q=QVEL(J)
      CALL INDVEL(Q,ANSA,ANST)
      UA(J)=ANSA
      UT(J)=ANST
    1 CONTINUE
C
C     COMPUTE UACHB(I),AND UTCHB(I),I=1(1)(IVELDG+1)
C
C     NA=LENGTH OF ARRAY A
C
      NA=50
      NLETH=51
      MCHEBY=JMAX
```

```
      NDEG=IVELDG
      CALL CHEBCF(UA,MCHEBY,NDEG,NLETH,A0,A,NA)
      UACHB(1)=A0
      DO 2 J=1,NDEG
      UACHB(J+1)=A(J)
    2 CONTINUE
      CALL CHEBCF(UT,MCHEBY,NDEG,NLETH,A0,A,NA)
      DO 3 J=1,NDEG
      UTCHB(J+1)=A(J)
    3 CONTINUE
      UTCHB(1)=A0
      RETURN
      END
C
C
C
C
      SUBROUTINE HATIJ
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS OF THE DOUBLE CHEBYSHEV
C     EXPANSION OF THE AXIAL AND TANGENTIAL INDUCTION
C     FACTORS, HA(I,J) AND HT(I,J) ,THAT IS,
C
C     IA(Q,QO)=SUMMATION'' HA(I,J)*T(I,Q) * T(J,QO),
C     I,J=0(1)IDEG
C
C     IT(Q,QO)=SUMMATION'' HT(I,J)*T(I,Q) * T(J,QO),
C     I,J=0(1)IDEG
C
C     DESCRIPTION OF PARAMETERS
C
C       QIND(I)     THE ITH ZERO OF T(ICHEBY+1,Q)
C
```

```
C                    QIND(I)=COS((2*I+1)*PAI/(2*(ICHEBY+1))
C                    I=0(1)ICHEBY
C
C
C
      COMMON /DATA3/IDEG,ICHEBY
      COMMON /DATA4/HA(31,31),HT(31,31)
      COMMON /DATA5/QIND(51)
      COMMON /DATA6/HUB
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /DATA9/EPSIND
      DIMENSION AXIAL(51,51),TANG(51,51)
      DIMENSION R(51),RO(51),RAMDA(51)
      DIMENSION PRJ(51,51),HAT(51,51)
C     COORDINATE TRANSFORMATION R=((1-HUB)*Q+1+HUB)/2
      IX=1+ICHEBY
      DO 1 I=1,IX
      R(I)=0.5 * ((1.-HUB)*QIND(I)+1.+HUB)
      RO(I)=R(I)
    1 CONTINUE
C     COMPUTE RAMDA AT R
      DO 2 I=1,IX
      Q=QIND(I)
      RAMDA(I)=RAMDAF(Q)
    2 CONTINUE
C     COMPUTE THE AXIAL AND TANGENTIAL INDUCTION
C     FACTORS
      NB=NBLADE
      NPR=IX
      NPRO=IX
C     INPUT DATA
C     KR,KRO: DIMENSION OF AXIAL,TANG
      KR=51
      KRO=51
      KCONT=KWENH
      EPSLON=EPSIND
```

```
      LUPER=LUPIND
      CALL INDFACT(NB,NPR,NPRO,R,RO,RAMDA,AXIAL,
     $ TANG,KR,KRO,KCONT,EPSLON,LUPER)
C     COMPUTE HA(I,J),AND HT(I,J)
      NC=KR
C
C     NA: THE LENGTH OF HA(NA,NA) AND HT(NA,NA)
      NCHEBY=ICHEBY
      NDEG=IDEG
      NAT=51
      NHAT=IDEG+1
      CALL DOUCHB(AXIAL,NCHEBY,NDEG,HAT,NAT,NC,PRJ,R)
      DO 3 I=1,NHAT
      DO 4 J=1,NHAT
      HA(I,J)=HAT(I,J)
    4 CONTINUE
    3 CONTINUE
      CALL DOUCHB(TANG,NCHEBY,NDEG,HAT,NAT,NC,PRJ,R)
      DO 5 I=1,NHAT
      DO 6 J=1,NHAT
      HT(I,J)=HAT(I,J)
    6 CONTINUE
    5 CONTINUE
      RETURN
      END
C
C
      FUNCTION RAMDAF(Q)
C
C
C     PURPOSE
C
C     TO COMPUTE RAMDA AT Q
C
C
C     THAT IS,
```

```
C
C     RAMDAF(Q)=AORAMD/2 + ARAMDA(J)* T(J,Q)  J=1,JRAMDA
C
      COMMON /DATA1/JRAMDA
      COMMON /DATA2/A0RAMD,ARAMDA(14)
      DIMENSION A(14)
      NLETH=14
      N=JRAMDA
      A0=A0RAMD
      DO 1 I=1,JRAMDA
      A(I)=ARAMDA(I)
    1 CONTINUE
      CALL SUMCHB(A0,A,NLETH,N,Q,ANS)
      RAMDAF=ANS
      RETURN
      END
C
C
      SUBROUTINE RAMCOE(ORAMDA,NOR)
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS OF THE CHEBYSHEV
C     EXPANSION OF RAMDA,THAT IS
C
C     RAMDA(Q)=A0RAMD/2 +ARAMDA(J)*T(J,Q)  J=1,JRAMDA
C
C     DESCRIPTION OF PARAMETERS
C
C
C       ORAMDA      VALUES OF RAMDA A ZEROS OF
C                   T(JRAMDA+1,Q)
C       NOR         LENGTH OF ORAMDA
      COMMON /CONST/PAI
      COMMON /DATA1/JRAMDA
```

```
      COMMON /DATA2/AORAMD,ARAMDA(14)
      M=JRAMDA
      H=PAI/FLOAT(2*(JRAMDA+1))
      FACT=2./FLOAT(JRAMDA+1)
      J=0
    1 IF(J.GT.JRAMDA) GO TO 2
      X=COS(FLOAT(J)*H)
      CALL SUMODD(ORAMDA,M,NOR~,X,ANS)
      IF(J.EQ.0) AORAMD=FACT*ANS
      IF(J.NE.0) ARAMDA(J)=FACT*ANS
      J=J+1
      GO TO 1
    2 RETURN
      END
C
C
      SUBROUTINE DOUCHB(C,NCHEBY,NDEG,A,NA,NC,PRJ,WK)
C
C
C     PURPOSE
C
C
C     TO COMPUTE THE COEFFICIENTS OF A FINITE
C     BIVARIATE CHEBYSHEV EXPANSION
C     THAT IS,
C
C     F(X,Y)=B(I,J) * T(I,X) * T(J,X), I,J=0(1)NDEG
C
C      WHERE  T(X,I)=CHEBYSHEV POLYNOMIAL  OF DEGREE I
C
C     B(0,0)=A(1,1)/4
C     B(I,0)=A(I+1,1)/2     (I .GT. 1)
C     B(0,J)=A(1,J+1)/2     (J .GT. 1)
C     B(I,J)=A(I+1,J+1)     (I,J .GT. 1)
C
C
```

```
C     DESCRIPTION OF PARAMETERS
C
C       NC          LENGTH OF C
C       NA          LENGTH OF A
C       C           FUNCTIONAL VALUES OF F(X,Y) AT THE
C                   DESCRETE  SET (NCHEBY+1) (NCHEBY+1) POINTS
C                   (X(I),Y(J)),    I,J=0 (1)NCHEBY
C                   X(I)=Y(I) THE ROOTS OF T(NCHEBY+1,X)
C                   X(I)=COS((2*I)*PAI/(28(NCHEBY+1)),
C                   I=0(1)NCHEBY
C       PRJ         WORK AREA (DIMENSION (NC,NA)
C       WK          WORK AREA  (DIMENSION (NC))
C
      DIMENSION C(NC,NC),A(NA,NA),PRJ(NC,NA),WK(NC)
      COMMON /CONST/PAI
      H=PAI/FLOAT(2*(NCHEBY+1))
      IRMAX=NCHEBY+1
      ISMAX=NCHEBY+1
      IMAX=NDEG+1
      JMAX=NDEG+1
C     COMPUTE PRJ
      DO 1 JX=1,JMAX
      J=JX-1
      X=COS(H*FLOAT(J))
      DO 2 IRX=1,IRMAX
C     CONSTRUCT WK
      DO 3  ISX=1,ISMAX
      WK(ISX)=C(IRX,ISX)
    3 CONTINUE
      M=NCHEBY
      CALL SUMODD(WK,M,NC,X,ANS)
      PRJ(IRX,JX)=ANS
    2 CONTINUE
    1 CONTINUE
C     COMPUTE COEFFICIENTS  A(I,J)
      FACT=4./((NCHEBY+1)**2)
```

```
      DO 4 IX=1,IMAX
      I=IX-1
      X=COS(H*FLOAT(I))
      DO 5  JX=1,JMAX
C     CONSTRUCT  WK
      DO 6 IRX=1,IRMAX
      WK(IRX)=PRJ(IRX,JX)
    6 CONTINUE
      M=NCHEBY
      CALL SUMODD(WK,M,NC,X,ANS)
      A(IX,JX)=FACT*ANS
    5 CONTINUE
    4 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CHEBCF(FTHI,MCHEBY,NDEG,NLETH,A0,A,NA)
C
C
C     PURPOSE
C
C
C     TO EVALUATE  THE COEFFICIENTS  OF THE CHEBYSHEV
C     EXPANSION OF THE FORM F(X)=A0/2 + A(K)  * T(K,X)
C     USING  CHEBYSHEV QUADRATURE
C
C     DESCRIPTION OF PARAMETERS
C
C       MCHEBY       THE NUMBER OF POINTS USED IN
C                    CHEBYSHEV QUADRATURE FORMULA
C       FTHI         FUNCTIONAL VALUES AT THE ROOTS OF
C                    T(MCHEBY,X),THI(I)=COS((2*I-1)*PAI/
C                    (2*MCHEBY)), I=1(1)MCHEBY
C       NDEG         DEGREE OF CHEBYSHEV  EXPANSION
C       NLETH        LENTH OF THI,ETA,FTHI,FETA
```

```
C       NA            LENGTH OF A
C       A0,A          COEFFICIENTS OF CHEBYSHEV EXPANSION
C
C
      DIMENSION A(NA),FTHI(NLETH)
      COMMON /CONST/PAI
      MODD=MCHEBY-1
      H=PAI/FLOAT(2*MCHEBY)
      K=0
    1 IF(K.GT.NDEG) GO TO 2
C     COMPUTE ALPHAK
      X=COS(H*FLOAT(K))
      CALL SUMODD(FTHI,MODD,NLETH,X,ANS)
      ALPHAK=2.*ANS/FLOAT(MCHEBY)
      IF(K.EQ.0) A0=ALPHAK
      IF(K.NE.0) A(K)=ALPHAK
      K=K+1
      GO TO 1
    2 RETURN
      END
C
C
      SUBROUTINE SUMODD(C,M,NLETH,X,ANS)
C
C
C     PURPOSE
C
C     TO COMPUTE  THE ODD POLYNOMIALS IN
C     CHEBYSHEV FORM AT X
C     THAT IS,
C
C     ANS=P(X)=C(J+1)*T(2*J+1,X) , J=0(1)M
C
C     DESCRIPTION OF PARAMETERS
C
C       NLETH      LENGTH OF C
```

```
C
      DIMENSION C(NLETH)
      K=M
      BK1=0.
      BK2=0.
      T=2.*X*X-1.
    1 BK=2.*T*BK1-BK2+C(K+1)
      IF(K.EQ.0) GO TO 3
      K=K-1
      BK2=BK1
      BK1=BK
      GO TO 1
    3 ANS=X*(BK-BK1)
      RETURN
      END
C
C
      SUBROUTINE SUMCHB(A0,A,NLETH,N,X,ANS)
C
C
C     PURPOSE
C
C     TO COMPUTE THE VALUE OF POLYNOMIALS IN
C     CHEBYSHEV FORMS
C     THAT IS,
C
C     ANS=P(X)=A0/2+A(K)*T(K,X), K=1(1)N
C
C     DESCRIPTION OF PARAMETERS
C
C         NLETH   LENGTH O  A
C
      DIMENSION A(NLETH)
      BK1=0.
      BK2=0.
      K=N
```

```
    1 IF(K.EQ.0) AK=A0
      IF(K.NE.0) AK=A(K)
      BK=2.*X*BK1-BK2+AK
      IF(K.EQ.0) GO TO 2
      K=K-1
      BK2=BK1
      BK1=BK
      GO TO 1
    2 ANS=0.5*(BK-BK2)
      RETURN
      END
C
C
      SUBROUTINE ZEROS(NDEGRE,NZERO,ZERO)
C
C
C     PURPOSE
C
C     TO COMPUTE THE ZEROS OF T(NDEGRE,Q)
C
C     DESCRIPTION OF PARAMETERS
C
C        ZERO(I)        ZERO(I)=COS((2*I-1)*PAI/(2*NDEGRE))
C                       I=1(1)NDEGRE
C
C        NZERO          LENGTH OF ZERO
C
C
      COMMON /CONST/PAI
      DIMENSION ZERO(NZERO)
      H=PAI/FLOAT(2*NDEGRE)
      DO 1 I=1,NDEGRE
      Z=H*FLOAT(2*I-1)
      ZERO(I)=COS(Z)
    1 CONTINUE
      RETURN
```

```
      END
C
C
C
      SUBROUTINE DOUSUM(A,B,M,N,NM,NN,X,Y,ANS)
C
C
C     PURPOSE
C
C     TO COMPUTE THE SUM OF A FINITE DOUBLE
C     CHEBYSHEV SERIES,THAT IS,
C
C     ANS=SUMMATION'' B(I,J)* T(I,X) * T(J,Y)
C
C     I=0(1)M, J=0(1)N
C
C     WHERE B(I,J)=A(I+1,J+1)
C
C     DESCRIPTION OF PARAMETERS
C
C       A       COEFFICIENTS OF THE DOUBLE
C               CHEBYSHEV SERIES
C       B       WORK AREA OF LENGTH NM
C       N       DEGREE IN X
C       M       DEGREE IN Y
C       NM      LENGTH OF THE ROW OF A
C       NN      LENGTH OF THE COLUMN OF A
C
C
      DIMENSION A(NM,NN),B(NM)
C     COMPUTE B(I)
      IMAX=M+1
      JMAX=N+1
      DO 1 IX=1,IMAX
      I=IX-1
      J=N
```

```
      DN1=0.
      DN2=0.
    2 DN=A(I+1,J+1)+2.*Y*DN1-DN2
      IF(J.EQ.0) GO TO 3
      J=J-1
      DN2=DN1
      DN1=DN
      GO TO 2
    3 B(I+1)=0.5 * (DN-DN2)
    1 CONTINUE
      I=M
      GM1=0.
      GM2=0.
    4 GM=B(I+1)+2.*X*GM1-GM2
      IF(I.EQ.0) GO TO 5
      I=I-1
      GM2=GM1
      GM1=GM
      GO TO 4
    5 ANS=0.5 * (GM-GM2)
      RETURN
      END
C
C
      SUBROUTINE FACTOR(Q,QO,AXIAL,TANG)
C
C     PURPOSE
C
C     TO COMPUTE THE AXIAL AND TANGENTIAL
C     INDUCTION FACTORS BY USING THE
C     DOUBLE CHEBYSHEV EXPANSIONS
C
C
      COMMON /DATA3/IDEG,ICHEBY
      COMMON /DATA4/HA(31,31),HT(31,31)
      DIMENSION A(31,31),B(31)
```

```
      NM=31
      NN=31
      M=IDEG
      N=IDEG
      IMAX=IDEG+1
      DO 1 I=1,IMAX
      DO 2 J=1,IMAX
      A(I,J)=HA(I,J)
    2 CONTINUE
    1 CONTINUE
      CALL DOUSUM(A,B,M,N,NM,NN,Q,QO,ANS)
      AXIAL=ANS
      DO 3 I=1,IMAX
      DO 4 J=1,IMAX
      A(I,J)=HT(I,J)
    4 CONTINUE
    3 CONTINUE
      CALL DOUSUM(A,B,M,N,NM,NN,Q,QO,ANS)
      TANG=ANS
      RETURN
      END
C
C
      FUNCTION CIRLF(Q)
C
C
C     PURPOSE
C
C     TO COMPUTE THE CIRCULATION AT Q
C
C     CIRLF(Q)=SQRT(1-Q**2) * SUMMATION G(M) * U(M-1,Q)
C
C
      COMMON /DATA11/MCIRCU
      COMMON /DATA14/G(15)
      N=MCIRCU-1
```

```
      BK1=0.
      BK2=0.
      K=N
    1 BK=G(K+1)+2.*Q*BK1-BK2
      IF(K.EQ.1) GO TO 2
      K=K-1
      BK2=BK1
      BK1=BK
      GO TO 1
    2 ANS=G(1)-BK1+2.*Q*BK
      CIRLF=SQRT(1.-Q*Q) * ANS
      RETURN
      END
C
C
      SUBROUTINE INDVEL(Q,UA,UT)
C
C
C     PURPOSE
C
C     TO COMPUTE THE AXIAL AND TANGENTIAL
C     COMPONENTS OF THE INDUCED VELOCITY
C
C
      COMMON /CONST/PAI
      COMMON /DATA3/IDEG,ICHEBY
      COMMON /DATA4/HA(31,31),HT(31,31)
      COMMON /DATA6/HUB
      COMMON /DATA11/MCIRCU
      COMMON /DATA14/G(15)
      DIMENSION TF(31),UF(31)
      IF(MCIRCU.GE.IDEG) IMAX=MCIRCU
      IF(MCIRCU.LE.IDEG) IMAX=IDEG
      NT=31
      NU=31
      CALL TCHEBY(TF,IMAX,Q,NT)
```

```
      CALL UCHEBY(UF,IMAX,Q,NU)
      SUMT=0.
      SUMA=0.
      DO 2 M=1,MCIRCU
      FM=G(M)*PAI*FLOAT(M)/(1.-HUB)
      I=0
      SUMAI=0.
      SUMTI=0.
    3 HI=1.
      IF(I.GT.IDEG) GO TO 4
      IF(I.EQ.0) HI=2.
      J=0
      SUMAJ=0.
      SUMTJ=0.
    5 HJ=1.
      IF(J.GT.IDEG) GO TO 6
      IF(J.EQ.0) HJ=2.
      IF(J.GT.M) GO TO 7
      SUMAJ=SUMAJ+HA(I+1,J+1)*TF(I+1)*TF(J+1)*
     $ UF(M)/(HI*HJ)
      SUMTJ=SUMTJ+HT(I+1,J+1)*TF(I+1)*TF(J+1)*
     $ UF(M)/(HI*HJ)
      J=J+1
      GO TO 5
    7 SUMAJ=SUMAJ+HA(I+1,J+1)*TF(I+1)*TF(M+1)*
     $ UF(J)/(HI*HJ)
      SUMTJ=SUMTJ+HT(I+1,J+1)*TF(I+1)*TF(M+1)*
     $ UF(J)/(HI*HJ)
      J=J+1
      GO TO 5
    6 SUMAI=SUMAI+SUMAJ
      SUMTI=SUMTI+SUMTJ
      I=I+1
      GO TO 3
    4 SUMA=SUMA+SUMAI*FM
      SUMT=SUMT+SUMTI*FM
```

```
    2 CONTINUE
      UA=SUMA
      UT=SUMT
      RETURN
      END
C
C
      SUBROUTINE VELOCIT(UA,UT,Q)
C
C     PURPOSE
C
C     TO INTERPOLATE THE INDUCED VELOCITY COMPONENTS
C     BY EVALUATING THE CHEBYSHEV EXPANSIONS OF
C     THE INDUCED VELOCITY COMPONENTS
C
      COMMON /DATA31/IVELDG,IVELCHB
      COMMON /DATA33/UACHB(51),UTCHB(51)
      DIMENSION A(50)
      NLETH=50
      N=IVELDG
      DO 1 I=1,N
      A(I)=UACHB(I+1)
    1 CONTINUE
      A0=UACHB(1)
      CALL SUMCHB(A0,A,NLETH,N,Q,ANS)
      UA=ANS
      DO 2 I=1,N
      A(I)=UTCHB(I+1)
    2 CONTINUE
      A0=UTCHB(1)
      CALL SUMCHB(A0,A,NLETH,N,Q,ANS)
      UT=ANS
      RETURN
      END
C
C
```

```
C
      SUBROUTINE TCHEBY(T,N,X,NT)
C
C
C     PURPOSE
C
C     TO COMPUTE THE VALUES OF CHEBYSHEV POLYNOMIALS
C     OF THE FIRST KIND OF DEGREES FROM 0-N AT X
C
C
      DOUBLE PRECISION Q,TN1,TN2,TN
      DIMENSION T(NT)
      Q=X
      TN2=1.D0
      TN1=Q
      T(1)=TN2
      T(2)=TN1
      IF(N.LE.1) RETURN
      IMAX=N+1
      DO 1 I=3,IMAX
      TN=2.D0 * Q * TN1 - TN2
      T(I)=TN
      TN2=TN1
      TN1=TN
    1 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE UCHEBY(U,N,X,NU)
C
C
C     PURPOSE
C
C     TO COMPUTE THE VALUES OF CHEBYSHEV POLYNOMIALS
C     OF THE SECOND KIND, U(I,X), I=0(1)N
```

```
C
C
      DOUBLE PRECISION Q,UN,UN1,UN2
      DIMENSION U(NU)
      Q=X
      UN2=1.D0
      UN1=2.D0 * Q
      U(1)=UN2
      U(2)=UN1
      IF(N.LE.1) RETURN
      IMAX=N+1
      DO 1 I=3,IMAX
      UN=2.D0 * Q* UN1 -UN2
      U(I)=UN
      UN2=UN1
      UN1=UN
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE INDFACT(NB,NPR,NPRO,R,RO,RAMDA,AXIAL,
     $ TANG,KR,KRO,KCONT,EPSLON,LUPER)
C
C     PURPOSE
C
C     TO COMPUTE  TABLES OF THE AXIAL AND TANGENTIAL
C     INDUCTION FACTORS
C
C
C     DATA FILE NEEDED
C
C     COSGRA
C
C     DESCRIPTION OF PARAMETERS
```

```
C
C        NB         NUMBER OF PROPELLER BLADES
C        NPR        NUMBER OF ELEMENTS OF ARRAY R
C        NPRO       NUMBER OF ELEMENTS OF ARRAY RO
C        KR         LENGTH OF ARRAY R
C        KRO        LENGTH OF ARRAY RO
C      AXIAL(K,I)   AXIAL INDUCTION FACTOR AT  (R(K),RO(I))
C      TANG(K,I)    TANGENTIAL INDUCTION FACTOR AT  (R(K),RO(I))
C      LUPER        UPPER LIMIT OF IA2
C      EPSLON       UPPER LIMIT OF IA3, EPSLON .GE. 0
C        R          ARRAY CONTAINING THE RADIAL COORDINATES
C                   OF REFERRENCE POINTS
C        RO         ARRAY CONTAINING THE RADIAL COORDINATES
C                   OF SOURCE POINTS
C                   R AND RO ARE ASSUMED TO BE EQUAL IF
C                   ABS(R-RO) .LE. 1.E-10
C      RAMDA(K)     HYDRODYNAMIC ADVANCE COEFFICIENT AT RO(K)
C      KCONT        CONTROL PARAMETER
C                   KCONT .EQ. 1 IF WRENCH MODIFIED FORMULA IS
C                   USED
C                   KCONT .NE. 1 IF ALTERNATE METHOD
C                   IS USED
C
C
      REAL R(KR),RO(KRO),AXIAL(KR,KRO),TANG(KR,KRO)
      REAL RAMDA(KRO)
      IF(KCONT .NE. 1) GO TO 5
C     COMPUTE INDUCTION FACTORS USING WRENCH'S
C     MODIFIED FORMULAS
      DO 6 K=1,NPRO
      ROU=RO(K)
      RAMDAK=RAMDA(K)
      DO 7 J=1,NPR
      RR=R(J)
      CALL WRENF(NB,RR,ROU,RAMDAK,ANSA,ANST)
      AXIAL(J,K)=ANSA
```

```
      TANG(J,K)=ANST
    7 CONTINUE
    6 CONTINUE
      RETURN
    5 DO 1 I=1,NPR
      DO 2 J=1,NPRO
      AXIAL(I,J)=0.
    2 CONTINUE
    1 CONTINUE
      CALL INDA12(EPSLON,NB,LUPER,NPR,NPRO,R,RO,RAMDA,
     $ AXIAL,KR,KRO)
      IF(EPSLON .LE. 1.E-10) GO TO 8
      CALL INDA3(EPSLON,NPR,NPRO,R,RO,RAMDA,
     $ AXIAL,KR,KRO)
    8 CALL INDA4(NPR,NPRO,R,RO,RAMDA,AXIAL,KR,KRO)
      B=FLOAT(NB)
      DO 3 K=1,NPRO
      ROU=RO(K)
      RAMDAK=RAMDA(K)
      DO 4 J=1,NPR
      RR=R(J)
      IF(ABS(RR-ROU) .LE. 1.E-10) AXIAL(J,K)=ROU/
     $ SQRT(RAMDAK*RAMDAK+RR*ROU)
      TANG(J,K)=RAMDAK*((ROU-RR)*B/RAMDAK-
     $ AXIAL(J,K))/RR
    4 CONTINUE
    3 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE INDA12(EPS,NB,LUPER,NPR,NPRO,R,RO,RAMDA,
     $ AXIAL,KR,KRO)
C
C
C
```

```
C     PURPOSE
C
C     TO COMPUTE IA12 BY USING A 25-POINT
C     GAUSS-LEGENDRE FORMULA ON EACH SUB-SUBINTERVAL
C
C
C     PRECISION: DOUBLE PRECISION
C
C
C
C
C     DESCRIPTION OF PARAMETERS
C
C       NBLADE   NUMBER OF PROPELLER BLADES-DEFINED IN
C                MAIN PROGRAM (USER SUPPLIED)
C       NLAST    NUMBER OF SUB-INTERVALS
C       IBEGIN   SUB-INTERVAL PARAMETER-DEFINED IN MAIN
C                PROGRAM (USER SUPPLIED)
C       IFINAL   SUB-INTERVAL PARAMETER-DEFINED IN MAIN
C                PROGRAM (USER SUPPLIED)
C                E.G.. IBEGIN=1, AND IFINAL=NLAST IF ALL
C                ALL SUN-INTERVALS ARE CONSIDERED
C       EPSLON   LOWER LIMIT OF INTEGRAL IA12
C        L       2 * PAI * L= UPER LIMIT OF INTEGRAL IA12
C                -DEFINED IN MAIN PROGRAM (SUER
C                SUPPLIED)
C       ALX      LOWER LIMIT OF EACH SUB-INTERVAL-
C                DEFINED IN MAIN PROGRAM (USER SUPPLIED)
C       AUX      UPER LIMIT OF EACH SUB-INTERVAL
C                -DEFINED IN MAIN PROGRAM (USER SUPPLIED)
C       NHN      NUMBER OF SUB-SUBINTERVALS OF EACH
C                SUB-INTERVAL (ALX(I),AUX(I))
C       NST      BLADE PARAMETER-DEFINED IN MAIN PROGRAM
C                (USER SUPPLIED)
C       NFI      BLADE PARAMETER-DEFINED IN MAIN PROGRAM
C                (USER SUPPLIED)
```

```
C               E.G. NST(I)=1 AND NFI(I)=1 IF ALL BLADES ARE
C               CONSIDERED IN SUB-INTERVAL I
C     ICHK      IF ICHK=1, THIS PROGRAM CALCULATES IA12 WHICH
C               HAS INDEX J.EQ.K AND (I-J).LE.1 AND PRINTS OUT
C               THESE RESULTS (SEE REMARK)
C
C
C    REMARK
C
C    THE PARAMETER NHN(I) FOR EACH SUB-INTERVAL MAY BE
C    CHOSEN AS FOLLOWS TO OBTAIN DESIRED NUMERICAL ACCURACY
C
C    STEP 1: SET ICHK=1
C    STEP 2: ASSUME NHN(I),I=1,NLAST
C    STEP 3: RUN THIS PROGRAM
C    STEP 4: INCREASE OR DECREASE NHN(I),I=1,NLAST
C    STEP 5: RUN THIS PROGRAM
C    STEP 6: COMPARE CURRENT OUTPUT WITH ALL PREVIOUS
C            OUTPUT
C    STEP 7: REPEAT STEPS 4 TO 6 UNTIL DESIRED ACCURACY
C            IS OBTAINED
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION ALX(20),AUX(20)
      REAL R(KR),RO(KRO),AXIAL(KR,KRO),RAMDA(KRO)
      REAL EPS
      INTEGER NST(20),NFI(20),NHN(20)
      COMMON /BLADE/CB(10),SB(10)
      COMMON /ADVAN/RAMDAK
      COMMON /COORD/RR,ROU
      COMMON /GQUZ5/NH,NSTART,NFINAL
      COMMON /GQUZ6/HLETH
      COMMON /GQUZ7/ZZ(25,5),CZZ(25,5),SZZ(25,5)
      EXTERNAL FAXIAL
      PAI=3.14159265358979323846264338327 9D0
```

```
      NBLADE=NB
C
C***  INPUT DATA
C
      ICHK=2
      NLAST=6
      EPSLON=EPS
      IBEGIN=1
      IFINAL=NLAST
      L=LUPER
      ALX(1)=EPSLON
      AUX(1)=PAI/100.D0
      NHN(1)=1
      NST(1)=1
      NFI(1)=NBLADE
      ALX(2)=AUX(1)
      AUX(2)=PAI/40.D0
      NHN(2)=1
      NST(2)=1
      NFI(2)=NBLADE
      ALX(3)=AUX(2)
      AUX(3)=PAI/4.D0
      NHN(3)=1
      NST(3)=1
      NFI(3)=NBLADE
      ALX(4)=AUX(3)
      AUX(4)=PAI
      NHN(4)=1
      NST(4)=1
      NFI(4)=NBLADE
      ALX(5)=AUX(4)
      AUX(5)=2.D0*PAI
      NHN(5)=1
      NST(5)=1
      NFI(5)=NBLADE
      ALX(6)=AUX(5)
```

```
      FLO=FLOAT(L)
      AUX(6)=2.D0*PAI*FLO
      NHN(6)=L-1
      NST(6)=1
      NFI(6)=NBLADE
C
C***  END OF INPUT
C
      FLO=FLOAT(NB)
      HH=2.D0 *PAI/FLO
      DO 20 I=1,NBLADE
      FLO=FLOAT(I-1)
      ZB=HH*FLO
      CB(I)=DCOS(ZB)
      SB(I)=DSIN(ZB)
   20 CONTINUE
      DO 105 INTEG=IBEGIN,IFINAL
      NSTART=NST(INTEG)
      NFINAL=NFI(INTEG)
      NH=NHN(INTEG)
      AL=ALX(INTEG)
      AU=AUX(INTEG)
      IF(ICHK .NE. 1) GO TO 100
      WRITE(6,60)
   60 FORMAT(5X,"INPUT DATA:"//)
      WRITE(6,61) AL
   61 FORMAT(5X,"LOWER LIMIT =",F25.15/)
      WRITE(6,62)AU
   62 FORMAT(5X,"UPER LIMIT =",F25.15/)
      WRITE(6,63) NB
   63 FORMAT(5X,"NUMBER OF BLADES:",I4/)
      WRITE(6,64)NSTART,NFINAL
   64 FORMAT(5X,"NSTART =",I2,2X,"NFINAL =",I3/)
      WRITE(6,65)NH
   65 FORMAT(5X,"NUMBER OF INTERVALS, NH=",I3/)
      WRITE(6,42)
```

```
   42 FORMAT(5X,"BLADE CONSIDERED :"/)
      WRITE(6,43) (I,I=NSTART,NFINAL)
   43 FORMAT(5X,"BLADE ",I2/)
      IF(ICHK.EQ.1) WRITE(6,26)
   26 FORMAT(5X,"NH",5X,"R",2X,"ROU",2X,"RAMDAK"/)
  100 CALL GQUDAD(AL,AU)
      DO 70 K=1,NPRO
      ROU=RO(K)
      RAMDAK=RAMDA(K)
      DO 71 J=1,NPR
      RR=R(J)
      ANS=0.D0
      IF(DABS(RR-ROU).LE.1.D-10) GO TO 72
      ANS=GQUZ25(FAXIAL)
   72 AXIAL(J,K)=AXIAL(J,K)+ANS
      IF(ICHK.EQ.1)WRITE(6,15) NH,RR,ROU,RAMDAK,ANS
   71 CONTINUE
   70 CONTINUE
   15 FORMAT(5X,I2,5X,F10.7,5X,F10.7,2X,F10.7,2X,D35.28/)
  105 CONTINUE
      RETURN
      END
C
C
      FUNCTION GQUZ25(AUX)
C
C     PURPOSE
C
C     TO COMPUTE  IA12 FOR A GIVEN INTERVAL BY USING
C     25-POINT GAUSS-LEGENDRE FORMULA
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION W(25)
      COMMON /GQUZ5/NH,NSTART,NFINAL
      COMMON /GQUZ6/HLETH
      COMMON /GQUZ7/ZZ(25,5),CZZ(25,5),SZZ(25,5)
```

```
      W(1)=0.11393798501026287947902964 1132D-1
      W(2)=0.26354986615032137261901815 2953D-1
      W(3)=0.40939156701306312655623487 7116D-1
      W(4)=0.54904695975835191925936891 5405D-1
      W(5)=0.68038333812356917207187185 6567D-1
      W(6)=0.80140700335001018013234959 6691D-1
      W(7)=0.91028261982963649811497220 7029D-1
      W(8)=0.10053594906705064420220689 0393D0
      W(9)=0.10851962447426365311609395 7050D0
      W(10)=0.11485825914571164833932554 5870D0
      W(11)=0.11945576353578477222817812 6513D0
      W(12)=0.12224244299031004168895951 8946D0
      W(13)=0.12317605372671545120390287 3079D0
      DO 1 I=14,25
      W(I)=W(26-I)
    1 CONTINUE
      H=HLETH
      NST=NSTART
      NFI=NFINAL
      SUM=0.D0
      DO 9 I=1,25
      R=0.D0
      DO 10 K=1,NH
      Z=ZZ(I,K)
      CZ=CZZ(I,K)
      SZ=SZZ(I,K)
      R=R+AUX(NST,NFI,Z,CZ,SZ)
   10 CONTINUE
      SUM=SUM+W(I)*R
    9 CONTINUE
      GQUZ25=0.5D0 *H*SUM
      RETURN
      END
C
C
      SUBROUTINE GQUDAD(A,B)
```

```
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS USED IN FUNCTION GQUZ25
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION T(25)
      COMMON /GQUZ5/NH,NSTART,NFINAL
      COMMON /GQUZ6/HLETH
      COMMON /GQUZ7/ZZ(25,5),CZZ(25,5),SZZ(25,5)
      T(1)=-0.995556969790498097908784946894D0
      T(2)=-0.976663921459517511498315386480D0
      T(3)=-0.942974571228974339414011169658D0
      T(4)=-0.894991997878275368851042006783D0
      T(5)=-0.833442628760834001421021108694D0
      T(6)=-0.759259263037357630577282865204D0
      T(7)=-0.673566368473468364485120633248D0
      T(8)=-0.577662930241222967723689841613D0
      T(9)=-0.473002731445714960522182115009D0
      T(10)=-0.361172305809387837735821730128D0
      T(11)=-0.243866883720988432045190362797D0
      T(12)=-0.122864692610710396387359818808D0
      T(13)=0.D0
      DO 1 I=14,25
      T(I)=-T(26-I)
    1 CONTINUE
      H=(B-A)/FLOAT(NH)
      HLETH=H
    5 FORMAT(2X,D31.24,2X,I2)
      DO 2 I=1,25
      DO 3 K=1,NH
      Z=A+.5*FLOAT(2*K-1)*H+.5*H*T(I)
      ZZ(I,K)=Z
      CZZ(I,K)=DCOS(Z)
      SZZ(I,K)=DSIN(Z)
```

```
    3  CONTINUE
    2  CONTINUE
       RETURN
       END
C
C
       FUNCTION FAXIAL(NST,NFI,Z,CZ,SZ)
C
C      PURPOSE
C
C      TO COMPUTE THE INTEGRAND OF IA12
C
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       COMMON /BLADE/CB(10),SB(10)
       COMMON /ADVAN/RAMDAK
       COMMON /COORD/RR,ROU
       SUM=0.D0
       DO 1 K=NST,NFI
       C=CZ*CB(K)-SZ*SB(K)
       A=(ROU-RR)*ROU*(ROU-RR*C)
       B=ROU*ROU+RR*RR-2.D0*ROU*RR*C+RAMDAK*
      $ RAMDAK*Z*Z
       SUM=SUM+A/DSQRT(B*B*B)
    1  CONTINUE
       FAXIAL=SUM
       RETURN
       END
C
C
       SUBROUTINE INDA3(EPS,NPR,NPRO,R,RO,RAMDA,
      $ AXIAL,KR,KRO)
C
C      PURPOSE
C
C      TO COMPUTE IA3
C
```

```
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL EPS,R(KR),RO(KRO),RAMDA(KR,KRO)
      REAL AXIAL(KR,KRO)
      COMMON /COORD/RR,ROU
      COMMON /ADVAN/RAMDAK
      EPSLON=EPS
      DO 1 K=1,NPRO
      ROU=RO(K)
      RAMDAK=RAMDA(K)
      DO 2 J=1,NPR
      RR=R(J)
      ANS=0.D0
      IF(DABS(RR-ROU) .LE. 1.D-10) GO TO 3
      ANS=AIND3(EPSLON)
    3 AXIAL(J,K)=AXIAL(J,K)+ANS
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
C
C
      FUNCTION AIND3(EPSLON)
C
C     PURPOSE
C
C     TO COMPUTE IA3
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /COORD/RR,ROU
      COMMON /ADVAN/RAMDAK
      A=(ROU-RR)*(ROU-RR)
      B=RR*ROU+RAMDAK*RAMDAK
      C=DSQRT(A+B*EPSLON*EPSLON)
      SUM=ROU*EPSLON/C
      AIND3=SUM
```

```
      IF(DABS(A).LE.1.D-15) RETURN
      F=0.5D0*ROU*RR*(ROU-RR)
      U=-EPSLON/(B*C)
      V=DLOG((EPSLON*DSQRT(B)+C)/DABS(ROU-RR))
      SUM=SUM+F*(U+V/DSQRT(B*B*B))
      AIND3=SUM
      RETURN
      END
C
C
      SUBROUTINE INDA4(NPR,NPRO,R,RO,RAMDA,AXIAL,KR,KRO)
C
C
C     PURPOSE
C
C     TO COMPUTE IA4
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL R(KR),RO(KRO),RAMDA(KRO),AXIAL(KR,KRO)
      COMMON /IND4/AN(15),C(15,15),COS3K(15,14),COS31
      COMMON /COORD/RR,ROU
      COMMON /ADVAN/RAMDAK
      REWIND1
      READ(1,1)NBLADE,NTERMS,LUPER
    1 FORMAT(3(2X,I3))
      PAI=3.14159265358979323846264338327 9D0
      READ(1,2)COS31
    2 FORMAT(2X,D21.14)
      DO 3 I=1,NTERMS
      KJ=I+1
      READ(1,4)(COS3K(I,J),J=1,KJ)
    3 CONTINUE
    4 FORMAT(5(2X,D21.14))
      DO 5 I=1,NTERMS
      READ(1,6)(C(I,J),J=1,I)
```

```
    5  CONTINUE
    6  FORMAT(3(2X,D35.28))
       L=LUPER
       FLO=FLOAT(L)
       UPER=2.D0*PAI*FLO
       NB=NBLADE
       CHK=100.D0
       DO 7 K=1,NPRO
       ROU=RO(K)
       RAMDAK=RAMDA(K)
       IF(DABS(RAMDAK-CHK).GE.1.D-8)CALL COEAN(UPER,NTERMS)
       DO 8 J=1,NPR
       RR=R(J)
       ANS=0.D0
       IF(DABS(RR-ROU).LE.1.D-10) GO TO 9
       ANS=AIND4(UPER,NTERMS,NB)
    9  AXIAL(J,K)=AXIAL(J,K)+ANS
    8  CONTINUE
    7  CONTINUE
       RETURN
       END
C
C
C
       FUNCTION AIND4(UPER,NTERMS,NB)
C
C
C      PURPOSE
C
C      TO COMPUTE IA4
C
C
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       REAL FLOAT
       COMMON /IND4/AN(15),C(15,15),COS3K(15,14),COS31
       COMMON /COORD/RR,ROU
```

```
      COMMON /ADVAN/RAMDAK
      U=UPER
      RAMDAO=RAMDAK
      X=RR
      XO=ROU
      P=0.25D0*(X*X+XO*XO)
      Q=-0.5D0*X*XO
      F=XO*U
      B=FLOAT(NB)
      CHK=0.D0
      SUM=F*(0.5D0*XO*B-X*COS31)/(RAMDAO*U)**3
      ANS=SUM*(XO-X)
      AIND4=ANS
      IF(NTERMS.LE.0) RETURN
      NMAX=NTERMS
      DO 1 N=1,NMAX
      T=F*P**(N) * (XO*B/FLOAT(2*N+2)-X*COS3K(N,1))
      KMAX=N
      DO 2 K=1,KMAX
      T=T+C(N,K)*P**(N-K)*Q**(K)*F*(XO*COS3K(N,K)-
     $ X*COS3K(N,K+1))
    2 CONTINUE
      T=T*AN(N)
      SUM=SUM+T
      IF(DABS(SUM-CHK).LE.1.D-16) GO TO 3
      CHK=SUM
    1 CONTINUE
    3 AIND4=SUM*(XO-X)
      RETURN
      END
C
C
C
      SUBROUTINE COEAN(UPER,NTERMS)
C
C
```

```
C     PURPOSE
C
C     TO COMPUTE AN(N)
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /IND4/AN(15),C(15,15),COS3K(15,14),COS31
      COMMON /ADVAN/RAMDAK
      RAMDAO=RAMDAK
      U=UPER
      NMAX=NTERMS
      A=DLOG(RAMDAO*U)
      SIGN=-1.
      R=2.*DLOG(2.D0)
      S=DLGAMA(1,1)
      DO 1 N=1,NMAX
      T=DLGAMA(N+1,1)+FLOAT(N)*R-(FLOAT(2*N+3)*
     $ A+S+DLGAMA(N+1,0))
      AN(N)=SIGN*DEXP(T)
      SIGN=-SIGN
    1 CONTINUE
      RETURN
      END
C
C
C
      FUNCTION DLGAMA(M,N)
C
C
C     PURPOSE
C
C
C     TO CALCULATE THE NATURAL LOGARITHM OF
C     GAMMA FUNCTION
C
C
```

```
C     DESCRIPTION OF PARAMETERS
C
C       M        INTEGER
C       N        CONTROL PARAMETER
C                    IF N=1, DLGAMA=LOG(GAMMA(M+1/2)
C                    IF N .NE. 1, DLGAMA=LOG(GAMMA(M))
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      X=FLOAT(M)
      IF(N .EQ. 1) GO TO 1
      IF(M .GE. 1) GO TO 2
      PRINT 3,M
    3 FORMAT(3X,"*** AN ERROR OCCURS IN DLGAMA
     $ DUE TO M=",I10,"***")
      STOP
    2 DLGAMA=0.D0
      IF(M.EQ.1) RETURN
      SUM=0.D0
      NMAX=M-1
      Z=0.D0
      DO 4 I=1,NMAX
      Z=Z+1.D0
      SUM=SUM+DLOG(Z)
    4 CONTINUE
      DLGAMA=SUM
      RETURN
    1 NMAX=M
      H=0.5D0 *1.144729885849400174143427351353D0
      SUM=0,D0
      Z=0.D0
      DO 5 I=1,NMAX
      Z=Z+1.
      SUM=SUM+DLOG(Z-0.5D0)
    5 CONTINUE
      DLGAMA=H+SUM
```

```
      RETURN
      END
C
C
C
      SUBROUTINE WRENF(NB,R,RO,RAMDAK,AXIAL,TANG)
C
C
C     PURPOSE
C
C     TO COMPUTE THE AXIAL AND TANGENTIAL
C     INDUCTION FACTORS AT POINT (R,RO) BY
C     USING WRENCH MODIFIED FORMULAS
C
C     DESCRIPTION OF PARAMETERS
C
C       NB      NUMBER OF PROPELLER BLADES
C       X       RADIAL COORDINATE OF THE REFERENCE POINT
C       XO      RADIAL COORDINATE OF THE SOURCE POINT
C       RAMDAO  HYDRODYNAMICAL ADVANCE COEFFICIENT AT XO
C       AXIAL   AXIAL INDUCTION FACTOR
C       TANG    TANGENIAL INDUCTION FACTOR
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL R,RO,RAMDAK,TANG,AXIAL
      X=R
      XO=RO
      RAMDAO=RAMDAK
      IF(DABS(X-XO).GT.1.D-10) GO TO 2
      U=DSQRT(RAMDAO*RAMDAO+X*XO)
      AXIAL=XO/U
      TANG=-RAMDAO/U
      RETURN
    2 Y=X/RAMDAO
      YO=XO/RAMDAO
```

```
      A=DSQRT(1.D0+Y**2)
      B=DSQRT(1.D0+YO**2)
      C=(YO*(A-1.D0)/(Y*(B-1.D0)))**(NB)
      D=(A-B)*FLOAT(NB)
      U=C*DEXP(D)
      A=DSQRT(DSQRT((1.D0+YO**2)/(1.D0+Y**2)))/
     $ (FLOAT(2*NB)*YO)
      B=(9.D0*YO**2+2.D0)/DSQRT((1.D0+YO**2)**3)
      C=(3.D0*Y**2-2.D0)/DSQRT((1.D0+Y**2)**3)
      D=(B+C)/FLOAT(24*NB)
      E=1.D0/(1.D0/U-1.D0)
      F=1.D0/(U-1.D0)
      IF(Y .LT. YO) F1=-A*(E+D*DLOG(1.D0+E))
      IF(Y .GT. YO) F2=A*(F-D*DLOG(1.D0+F))
      IF(Y.GT.YO) GO TO 1
      FLO=FLOAT(NB)
      AXIAL = FLO*YO*(1.D0-Y/YO)*(1.D0-2.D0*FLO*
     $ YO*F1)
      TANG=-2.D0*FLO*FLO*YO*(1.D0-YO/Y)*F1
      RETURN
    1 FLO=FLOAT(NB)
      AXIAL=2.D0*FLO*FLO*YO*Y*(1.D0-YO/Y)*F2
      TANG=-FLO*(1.D0-YO/Y)*(1.D0+2.D0*FLO*YO*F2)
      RETURN
      END
C
C     ***********************************
C     *****                         *****
C     *****    ACOUSTIC MODEL       *****
C     *****                         *****
C     ***********************************
C
C
      SUBROUTINE MEANSQ
C
C     PURPOSE
```

```
C
C     TO COMPUTE THE ENSEMBLE MEAN SQUARE  OF
C     THE PROPELLER ACOUSTIC NOISE DUE TO
C     ROTATIONAL FORCES
C
C     WEIGHTS OF A 25-POINT GAUSS-LEGENDRE
C     FORMULA
C
      COMMON /DATA24/SQMEAN
      COMMON /ACOUST/PRESU(25)
      DOUBLE PRECISION W(13)
      W(1)=0.11393798501026287947902964113 2D-1
      W(2)=0.26354986615032137261901815295 3D-1
      W(3)=0.40939156701306312655623487711 6D-1
      W(4)=0.54904695975835191925936891540 5D-1
      W(5)=0.68038333812356917207187185656 7D-1
      W(6)=0.80140700335001018013234959669 1D-1
      W(7)=0.91028261982963649811497220702 9D-1
      W(8)=0.10053594906705064420220689039 3D0
      W(9)=0.10851962447426365311609395705 0D0
      W(10)=0.11485825914571164833932554587 0D0
      W(11)=0.11945576353578477222817812651 3D0
      W(12)=0.12224244299031004168895951894 6D0
      W(13)=0.12317605372671545120390287307 9D0
      CALL PRESUF
      SUM=0.
      DO 1 I=1,25
      Z=PRESU(I)
      IF(I.LE.13)  SUM=SUM+W(I)*Z*Z
      IF(I.GT.13)  SUM=SUM+W(26-I)*Z*Z
    1 CONTINUE
      SQMEAN=SUM*0.5
      RETURN
      END
C
C
```

```
      SUBROUTINE PRESUF
C
C     PURPOSE
C
C     TO COMPUTE THE PROPELLER ACOUSTIC
C     PRESSURE
C
C
      COMMON /CONST/PAI
      COMMON /DATA6/HUB
      COMMON /DATA11/MCIRCU
      COMMON /DATA14/G(15)
      COMMON /DATA20/NNOISE,NOSCHB
      COMMON /DATA22/PK0(25,31),PKA(25,31),PKT(25,31
      COMMON /DATA31/IVELDG,IVELCHB
      COMMON /DATA33/UACHB(51),UTCHB(51)
      COMMON /ACOUST/PRESU(25)
      DO 1 I=1,25
      PRESU(I)=0.
    1 CONTINUE
      FACT=(1.-HUB)*PAI/8.
      DO 2 M=1,MCIRCU
      IF((M-1).GT.NNOISE) GO TO 3
      IF((M+1).GT.NNOISE) GO TO 4
      DO 5 I=1,25
      PRESU(I)=PRESU(I)+FACT*G(M)*(PK0(I,M)-
     $ PK0(I,M+2))
    5 CONTINUE
      GO TO 2
    4 DO 6 I=1,25
      PRESU(I)=PRESU(I)+FACT*G(M)*PK0(I,M)
    6 CONTINUE
    2 CONTINUE
    3 FACT=(1.-HUB)*PAI/16.
      DO 8 I=1,25
      PRESU(I)=PRESU(I)+0.25*FACT*G(1)*(PKA(I,1)*
```

```
     $ UACHB(1)+PKT(I,1)*UTCHB(1))
   8  CONTINUE
      WA=FACT*UACHB(1)/2.
      WT=FACT*UTCHB(1)/2.
      DO 9 M=1,MCIRCU
      IF(M.NE.1) GO TO 10
      DO 11 I=1,25
      PRESU(I)=PRESU(I)+G(M)*(WA*(PKA(I,M)/2.-
     $ PKA(I,M+2))+WT*(PKT(I,M)/2.-PKT(I,M+2)))
  11  CONTINUE
      GO TO 9
  10  IF((M-1).GT.NNOISE) GO TO 12
      IF((M+1).GT. NNOISE) GO TO 13
      DO 14 I=1,25
      PRESU(I)=PRESU(I)+G(M)*(WA*(PKA(I,M)-
     $ PKA(I,M+2))+WT*(PKT(I,M)-PKT(I,M+2)))
  14  CONTINUE
      GO TO 9
  13  DO 15 I=1,25
      PRESU(I)=PRESU(I)+G(M)*(WA*PKA(I,M)+WT*PKT(I,M))
  15  CONTINUE
   9  CONTINUE
  12  KEND=NNOISE+1
      DO 16 M=1,MCIRCU
      DO 17 KX=1,KEND
      K=KX-1
      HK=1.
      IF(K.EQ.0) HK=2.
      IF((K+M-1).GT.IVELDG) GO TO 24
      HAT=1.
      IF((K+M-1).EQ.0) HAT=2.
      DO 18 I=1,25
      PRESU(I)=PRESU(I)+FACT*G(M)*(UACHB(K+M)*
     $ PKA(I,K+1)+UTCHB(K+M)*PKT(I,K+1))/(HK*HAT)
  18  CONTINUE
  24  IA=IABS(K-M+1)
```

```
      IF(IA.GT.IVELDG) GO TO 19
      HAT=1.
      IF(IA.EQ.0) HAT=2.
      DO 20 I=1,25
      PRESU(I)=PRESU(I)+FACT*G(M)*(UACHB(IA+1)*
     $ PKA(I,K+1)+UTCHB(IA+1)*PKT(I,K+1))/(HAT*HK)
   20 CONTINUE
   19 IF((M+K+1).GT.IVELDG) GO TO 21
      DO 22 I=1,25
      PRESU(I)=PRESU(I)-FACT*G(M)*(UACHB(M+K+2)*
     $ PKA(I,K+1)+UTCHB(M+K+2)*PKT(I,K+1))/HK
   22 CONTINUE
   21 IA=IABS(K-M-1)
      IF(IA.GT.IVELDG) GO TO 17
      HAT=1.
      IF(IA.EQ.0) HAT=2.
      DO 23 I=1,25
      PRESU(I)=PRESU(I)-FACT*G(M)*(UACHB(IA+1)
     $ *PKA(I,K+1)+UTCHB(IA+1)*PKT(I,K+1))/(HAT*HK)
   23 CONTINUE
   17 CONTINUE
   16 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PK0AT
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS OF THE
C     CHEBYSHEV EXPANSIONS OF K0,KA, AND KT
C
C
      COMMON /DATA6/HUB
```

```
      COMMON /DATA19/QNOISE(51)
      COMMON /DATA20/NNOISE,NOSCHB
      COMMON /DATA21/ETA(25)
      COMMON /DATA22/PKO(25,31),PKA(25,31),PKT(25,31)
      COMMON /CONST/PAI
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      DIMENSION AO(51),AA(51),AT(51),RD(51)
      NRD=51
      JX=NOSCHB+1
      NB=NBLADE
      XB=FLOAT(NB)
      HB=2.*PAI/XB
      HI=PAI/XB
C     COMPUTE PKO(I,J), PKA(I,J), AND PKT(I,J),
C     I=1(1)25, J=1(1)NNOISE+1
      DO 1 I=1,25
C     COORDINATE TRANSFORMATION
      PHIO=HI*(1.+ETA(I))
C     COMPUTE THE FUNCTIONAL VALUES OF PKO,
C     PKA, AND PKT
      DO 2 J=1,JX
      AO(J)=0.
      AA(J)=0.
      AT(J)=0.
    2 CONTINUE
      DO 3 K=1,NBLADE
      DELTAK=HB*FLOAT(K-1)
C     CALL SUBROUTINE RETARD TO GET THE
C     RETARDED DISTANCE
C
      CALL RETARD(PHIO,DELTAK,RD,NRD)
      DO 4 J=1,JX
      RETA=RD(J)
      ROU=0.5*((1.-HUB)*QNOISE(J)+1.+HUB)
      CALL FKOAT(RETA,PHIO,DELTAK,ROU,ANSO,ANSA,ANST)
      AO(J)=AO(J)+ANSO
```

```
      AA(J)=AA(J)+ANSA
      AT(J)=AT(J)+ANST
    4 CONTINUE
    3 CONTINUE
C     COMPUTE THE COEFFICIENTS OF KO, KA, AND KT
      MCHEBY=NOSCHB+1
      NDEG=NNOISE
      NA=51
      NLETH=51
      JMAX=NNOISE+1
      CALL CHEBCF(AO,MCHEBY,NDEG,NLETH,RO,RD,NRD)
      DO 5 J=1,JMAX
      IF(J.EQ.1) PKO(I,J)=RO
      IF(J.NE.1) PKO(I,J)=RD(J-1)
    5 CONTINUE
      CALL CHEBCF(AA,MCHEBY,NDEG,NLETH,RO,RD,NRD)
      DO 6 J=1,JMAX
      IF(J.EQ.1) PKA(I,J)=RO
      IF(J.NE.1) PKA(I,J)=RD(J-1)
    6 CONTINUE
      CALL CHEBCF(AT,MCHEBY,NDEG,NLETH,RO,RD,NRD)
      DO 7 J=1,JMAX
      IF(J.EQ.1) PKT(I,J)=RO
      IF(J.NE.1) PKT(I,J)=RD(J-1)
    7 CONTINUE
    1 CONTINUE
      REWIND2
      DO 8 I=1,25
      DO 9 J=1,JMAX
      WRITE(2,10) PKO(I,J),PKA(I,J),PKT(I,J)
    9 CONTINUE
    8 CONTINUE
   10 FORMAT(3(2X,F23.16))
   20 FORMAT(//,5X,"*******************************"//)
   21 FORMAT(5X,"  REPLACE FILE PKOATD !?"/)
   22 FORMAT(5X,"*******************************"//)
```

```
      WRITE(6,20)
      WRITE(6,21)
      WRITE(6,22)
      RETURN
      END
C
C
      SUBROUTINE FKOAT(RETA,PHIO,DELTAK,ROU,
     $ ANSO,ANSA,ANST)
C
C     PURPOSE
C
C     TO COMPUTE KO,KA, AND KT
C
C
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA17/XDISTA,AZIMUT,ANGLE,SANGLE,CANGLE
      SI=SIN(PHIO+DELTAK+TM*RETA)
      CI=COS(PHIO+DELTAK+TM*RETA)
      C1=ROU/RETA-(XDISTA/RETA)*SANGLE*CI
      C2=(XDISTA/RETA)*CANGLE -FM
      C3=(XDISTA/RETA)*SANGLE*SI
      RM=-FM*C2+ROU*TM*C3
      DRM=ROU*TM*C1
      SQM=FM*FM+(ROU*TM*ROU*TM)
      CPLUS=1.-RM
      C4=(1.-SQM)/RETA+TM*DRM
      C5=0.5/(XDISTA*CPLUS*CPLUS)
      ANSO=C5*(TM*VF*C1+(C4/CPLUS)*(ROU*C2/
     $ RAMDAP+VF*C3))
      ANSA=C5*(-XDISTA*TM*SANGLE*CI/RETA+
     $ (C4/CPLUS)*C3)
      ANST=C5*(FM/RETA+(C4/CPLUS)*C2)
      RETURN
      END
```

```
C
C
      SUBROUTINE RETARD(PHIO,DELTAK,RD,NRD)
C
C
C     PURPOSE
C
C     TO COMPUTE THE RETARDED DISTANCE
C
C
      DOUBLE PRECISION C1,C2,C3,C4,C5,C6,C7
      DOUBLE PRECISION C8,C9,TRY,FEWTON
      DIMENSION RD(NRD)
      COMMON /DATA6/HUB
      COMMON /DATA8/NBLADE,KWENH,LUPIND
      COMMON /DATA12/FM,TM,VF,RAMDAP
      COMMON /DATA17/XDISTA,AZIMUT,ANGLE,SANGLE,CANGLE
      COMMON /DATA19/QNOISE(51)
      COMMON /DATA20/NNOISE,NOSCHB
      COMMON /RETIME/C1,C2,C3,C4,C5,C6,C7,C8,C9
      C1=1.D0-FM*FM
      C2=FM*CANGLE
      C3=C2*C2
      C7=PHIO+DELTAK
      C8=1.D0/XDISTA
      C9=TM
      IX=NOSCHB+1
      TRY=XDISTA*(-C2+DSQRT(C3+C1))/C1
      DO 1 I=1,IX
      R=0.5D0*((1.D0-HUB)*QNOISE(I)+1.D0+HUB)
      C4=1.D0+R*R/(XDISTA*XDISTA)
      C5=2.*R*SANGLE/XDISTA
      C6=TM*R*SANGLE/XDISTA
      RD(I)=FEWTON(TRY)
      TRY=RD(I)
    1 CONTINUE
```

```
      RETURN
      END
C
C
      FUNCTION FEWTON(TRY)
C
C
C     PURPOSE
C
C     TO COMPUTE THE RETARDED DISTANCE BY USING
C     NEWTON METHOD
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /RETIME/C1,C2,C3,C4,C5,C6,C7,C8,C9
      EPSLON=1.D-16
      RN1=TRY
    1 RN=RN1
      C=DCOS(C7+C9*RN)
      S=DSIN(C7+C9*RN)
      SQT=DSQRT(C3+C1*(C4-C5*C))
      U=RN*C8-(-C2+SQT)/C1
      V=C8-C6*S/SQT
      RN1=RN-U/V
      IF(DABS(RN1-RN) .GT. EPSLON) GO TO 1
      FEWTON=RN1
      RETURN
      END
C
C
      SUBROUTINE ABSIAS
C
C
C     PURPOSE
C
C     TO SUPPLY THE ABSCISSAS OF
```

```
C     A 25-POINT GAUSS-LEGENDRE FORMULA
C
      DOUBLE PRECISION T(13),W(13)
      COMMON /DATA21/ETA(25)
      W(1)=0.11393798501026287947902964113 2D-1
      W(2)=0.26354986615032137261901815295 3D-1
      W(3)=0.40939156701306312655623487711 6D-1
      W(4)=0.54904695975835191925936891540 5D-1
      W(5)=0.68038333812356917207187185656 7D-1
      W(6)=0.80140700335001018013234959669 1D-1
      W(7)=0.91028261982963649811497220702 9D-1
      W(8)=0.10053594906705064420220689039 3D0
      W(9)=0.10851962447426365311609395705 0D0
      W(10)=0.11485825914571164833932554587 0D0
      W(11)=0.11945576353578477222817812651 3D0
      W(12)=0.12224244299031004168895951894 6D0
      W(13)=0.12317605372671545120390287307 9D0
      T(1)=-0.99555696979049809790878494689 4D0
      T(2)=-0.97666392145951751149831538648 0D0
      T(3)=-0.94297457122897433941401116965 8D0
      T(4)=-0.89499199787827536885104200678 3D0
      T(5)=-0.83344262876083400142102110869 4D0
      T(6)=-0.75925926303735763057728286520 4D0
      T(7)=-0.67356636847346836448512063324 8D0
      T(8)=-0.57766293024122296772368984161 3D0
      T(9)=-0.47300273144571496052218211500 9D0
      T(10)=-0.36117230580938783773582173012 8D0
      T(11)=-0.24386688372098843204519036279 7D0
      T(12)=-0.12286469261071039638735981880 8D0
      T(13)=0.D0
      DO 1 I=1,25
      IF(I.LE.13) ETA(I)=T(I)
      IF(I.GT.13) ETA(I)=-T(26-I)
    1 CONTINUE
      RETURN
      END
```

```
      PROGRAM COECOS(INPUT,OUTPUT,COSGRA,TAPE4=COSGRA,
     $ TAPE6=OUTPUT)
C
C
C     PURPOSE
C
C     TO COMPUTE THE COEFFICIENTS NEEDED IN
C     CALCULATING THE LIFTING-LINE INDUCTION
C     FACTORS BY MEANS OF THE ALTERNATE METHOD
C     DECELOPED IN APPENDIX A
C
C     THESE COEFFICIENTS ARE:
C
C     COS31=INTEGRAL OF (G(X,0)/X**3)*DX (FROM 1 TO INFINTY)
C
C     COS3K(N,K)=INTEGRAL OF (G(X,K)/X**(2*N+3))*DX
C                (FROM 1 TO INFINTY)
C
C     C(I,J)=(I!)/((I-J)! * (J!))
C
C     WHERE
C           G(X,K)=SUMMATION OF COS(Y(I))**K, I=1,2,..NB
C           NB=NUMBER OF PROPELLER BLADES
C           Y(I)=X*U+(2*PAI/NB)*(I-1)
C
C
C     PRECISION: DOUBLE PRECISION
C
C
C     DESCRIPTION OF PARAMETERS
C
C      NBLADE  NUMBER OF PROPELLER BLADES-DEFINED IN MAIN
C              PROGRAM (INPUT)
C      NTERMS  NUMBER OF TERMS DESIRED IN CALCULATING IA4
C              DEFINED IN MAIN PROGRAM (INPUT)
```

```
C         L     U=2 * PAI * L  LOWER LIMIT OF
C                 INTEGRAL IA4  (INPUT)
C         C     C(I,J)=(I!)/((I-J)! * (J!))
C
C
C     ALL OUTPUTS ARE WRITTEN ONTO FILE COSGRA
C
C     COSGRA  CONTAINS :
C
C                         NBLADE
C                         NTERMS
C                         COS31
C                         COS3K(N,K)
C                         C(N,I)
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /BBB/C(15,15)
      COMMON /CCC/COS3K(15,16),COS31
      REWIND4
      PAI=3.14159265358979323846264338327900
C
C***  INPUT DATA
C
      NBLADE=4
      NTERMS=12
      L=4
C
C***  END OF INPUT
C
      NB=NBLADE
      XL=FLOAT(L)
      U=2.D0*PAI*XL
      WRITE(6,20)
   20 FORMAT(5X,"INPUT DATA"//)
      WRITE(6,21)NB
```

```
21   FORMAT(5X,"NUMBER OF BLADES: ",I3/)
     WRITE(6,22)
22   FORMAT(5X,"NUMBER OF TERMS USED IN THE POWER SERIES"/)
     WRITE(6,23) NTERMS
23   FORMAT(5X,"NTERMS =",I3/)
     WRITE(6,24)
24   FORMAT(5X,"THE LOWER LIMIT = 2* PAI * L"/)
     WRITE(6,25) L
25   FORMAT(5X,"WHERE L ="I3//)
     CALL COECNK(NTERMS)
     CALL COESCF(NB,NTERMS,U)
     WRITE(4,6)NB,NTERMS,L
6    FORMAT(3(2X,I3))
     CCOS31=COS31
     WRITE(4,4)COS31
4    FORMAT(2X,D21.14)
     DO 1 I=1,NTERMS
     KJ=I+1
     WRITE(4,5)  (COS3K(I,J),J=1,KJ)
1    CONTINUE
5    FORMAT(5(2X,D21.14))
     DO 2 I=1,NTERMS
     WRITE(4,7) (C(I,J),J=1,I)
7    FORMAT(3(2X,D35.28))
2    CONTINUE
     ENDFILE4
     WRITE(6,77)
77   FORMAT(///)
     WRITE(6,8)
     WRITE(6,9)
     WRITE(6,10)
     WRITE(6,9)
     WRITE(6,8)
8    FORMAT(5X,"*************************************")
9    FORMAT(5X,"*****                           *****")
10   FORMAT(5X,"*****     REPLACE COSGRA        *****")
```

```
      STOP
      END
C
C
      SUBROUTINE COESCF(NB,NTERMS,U)
C
C     PURPOSE
C
C     TO COMPUTE COS31 AND COS3K
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /CCC/COS3K(15,16),COS31
      NMAX=NTERMS
      CALL SCNK(NB,U,1,3,ANS)
      COS31=ANS
      DO 1 N=1,NMAX
      KMAX=N+1
      NI=2*N+3
      DO 2 K=1,KMAX
      KI=K
      CALL SCNK(NB,U,KI,NI,ANS)
      COS3K(N,K)=ANS
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SCNK(NB,U,KI,NI,ANS)
C
C     PURPOSE
C
C     TO COMPUTE THE FOLLOWING INTEGRAL
C
C     ANS=INTEGRAL OF (G(X,KI)/X**NI)*DX (FROM 1
C         TO INFINTY)
```

```
C
C     WHERE
C           G(X,KI)= SUMMATION OF (COS(Y(I)**KI),I=1,..,NB
C           Y(I)= U*X+2*PAI*(I-1)/NB
C           U= THE LOWER LIMIT OF  IA4
C           NB= THE NUMBER OF PROPELLER BLADES
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL FLOAT
      ANS=0.D0
      IF(NI .LE. 1) RETURN
      KT=1
      NT=1
      IF(((KI/2)*2-KI).EQ.0) KT=2
      IF(((NI/2)*2-NI) .EQ.0) NT=2
      K=KI/2
      N=NI/2
      IF(KT.EQ.1) K=(KI+1)/2
      IF(NT.EQ.1) N=(NI+1)/2
      IF(KT .EQ. 2) GO TO 2
      S=DLGAMA(2*K,0)
      SUM=0.D0
      DO 3 J=1,K
      IF((((2*J-1)/NB)*NB-(2*J-1)) .NE. 0) GO TO 3
      A=U*FLOAT(2*J-1)
      CALL SINCOS(N,A,SINE,SINO,COSE,COSO)
      IF(NT .EQ. 1) Z=COSO
      IF(NT .EQ. 2) Z=COSE
      SUM=SUM+Z*DEXP(S-DLGAMA(K-J+1,0)-DLGAMA(K+J,0))
    3 CONTINUE
      FLO=FLOAT(NB)
      ANS= FLO*(SUM/((2.D0)**(2*K-2)))
      RETURN
    2 S=DLGAMA(2*K+1,0)
      SUM=0.D0
      DO 4 J=1,K
```

```
      IF(((2*J/NB)*NB-2*J) .NE. 0) GO TO 4
      FLO=FLOAT(2*J)
      A= FLO*U
      CALL SINCOS(N,A,SINE,SINO,COSE,COSO)
      IF(NT .EQ. 1) Z=COSO
      IF(NT .EQ. 2 ) Z=COSE
      SUM=SUM+Z*DEXP(S-DLGAMA(K-J+1,0)-DLGAMA(K+J+1,0))
    4 CONTINUE
      FLO=FLOAT(NI-1)
      ANS=DEXP(S-2.D0 *DLGAMA(K+1,0))*.5D0 /
     $ FLO+SUM
      FLO=FLOAT(NB)
      ANS=ANS*FLO/((2.D0)**(2*K-1))
      RETURN
      END
C
C
      SUBROUTINE SINCOS(N,A,SINE,SINO,COSE,COSO)
C
C     PURPOSE
C
C     TO CALCULATE THE FOLLOWING INTEGRALS FOR
C     N .GE.1 AND U .GT. 40
C
C     SINE(N,A)=INTEGRAL OF (SIN(A*X)/X**(2*N))*DX
C     COSE(N,A)=INTEGRAL OF (COS(A*X)/X**(2*N))*DX
C     SINO(N,A)=INTEGRAL OF (SIN(A*X)**(2*N-1))*DX
C     COSO(N,A)=INTEGRAL OF (COS(A*X)**(2*N-1))*DX
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL FLOAT
      EPSLON=1.D-15
      SUMR=1.D0
      SUMS=1.D0
      SUMT=1.D0
      CHKR=0.D0
```

```
      CHKS=0.D0
      CHKT=0.D0
      FR=1.D0
      FS=1.D0
      FT=1.D0
      KR=0
      KS=0
      KT=0
      X= FLOAT(2*N-1)
      Y=X+1.
1     IF(KT.EQ.0) FT=-FT*X*Y/A**2
      X=Y
      Y=X+1.D0
      IF(KR.EQ.0) FR=-FR*X*Y/A**2
      X=Y
      Y=X+1.
      IF(KS.EQ.0) FS=-FS*X*Y/A**2
      IF(KR.EQ.0) SUMR=SUMR+FR
      IF(KS.EQ.0) SUMS=SUMS+FS
      IF(KT.EQ.0) SUMT=SUMT+FT
      IF(DABS(SUMT-CHKT)/DABS(SUMT) .LE. EPSLON) KT=1
      IF(DABS(SUMR-CHKR)/DABS(SUMR) .LE. EPSLON) KR=1
      IF(DABS(SUMS-CHKS)/DABS(SUMS) .LE. EPSLON) KS=1
      IF((KR+KS+KT-3) .EQ. 0) GO TO 2
      CHKR=SUMR
      CHKS=SUMS
      CHKT=SUMT
      GO TO 1
2     R=SUMR/A
      FL=FLOAT(2*N)
      S=SUMS* FL/A**2
      T=SUMT/A
      FL=FLOAT(2*N-1)
      U=SUMR* FL/A**2
      SI=DSIN(A)
      CO=DCOS(A)
```

```
      SINE=R*CO+S*SI
      SINO=T*CO+U*SI
      COSE=S*CO-R*SI
      COSO=U*CO-T*SI
      RETURN
      END
C
C
      SUBROUTINE COECNK(NMAX)
C
C     PURPOSE
C
C     TO COMPUTE C(N,K)
C     WHERE C(N,K)=N!/((N-K)! * K!)
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /BBB/C(15,15)
      IF(NMAX.EQ.0) RETURN
      DO 1 N=1,NMAX
      A=DLGAMA(N+1,0)
      KMAX=N
      DO 2 K=1,KMAX
      C(N,K)=DEXP(A-DLGAMA(K+1,0) -DLGAMA(N-K+1,0))
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
C
C
      FUNCTION DLGAMA(M,N)
C
C     PURPOSE
C
C     TO CALCULATE THE NATURAL LOGARITHM OF GAMMA
C     FUNCTION
C     WHERE M IS AN INTEGER, N A CONTROL PARAMETER
```

```
C
C       DLGAMA=LOG(GAMMA(M+1/2)) IF N=1
C       DLGAMA=LOG(GAMMA(M)) IF N.NE.1
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        X=FLOAT(M)
        IF(N .EQ. 1) GO TO 1
        IF(M .GE. 1) GO TO 2
        PRINT 3, M
  3     FORMAT(3X,"*** AN ERROR OCCURS IN
       $ DLGAMA DUE TO M=",I10,"***")
        STOP
  2     DLGAMA=0.D0
        IF(M.EQ.1) RETURN
        SUM=0.D0
        NMAX=M-1
        Z=0.D0
        DO 4 I=1,NMAX
        Z=Z+1.D0
        SUM=SUM+DLOG(Z)
  4     CONTINUE
        DLGAMA=SUM
        RETURN
C       N=1
  1     NMAX=M
C       H=LOG(SQRT(PAI))=LOG(GAMMA(1/2))
        H=.5D0 *1.1447298858494001741434273513530D0
        SUM=0.D0
        Z=0.D0
        DO 5 I=1,NMAX
        Z=Z+1.D0
        SUM=SUM+DLOG(Z-0.5D0)
  5     CONTINUE
        DLGAMA=H+SUM
        RETURN
        END
```